

Lezione 05

Dott. Francesco De Angelis

francesco.deangelis@unicam.it

<http://francescodeangelis.org>



Scuola di Scienze e Tecnologie - Sezione di Informatica
Università di Camerino

Agenda

- Le reti di computer: questioni tecniche [cap. 17]
- Organizzare le informazioni: liste, query, markup [cap. 18]
- Il Sistema Operativo [cap. 19]
- Al lavoro sui documenti [cap. 20]



Informatica di base 5e

Dennis P. Curtin, Kim Foley, Kunal Sen, Cathleen Morin.

A cura di Agostino Marengo

Copyright © 2012 – The McGraw-Hill Companies

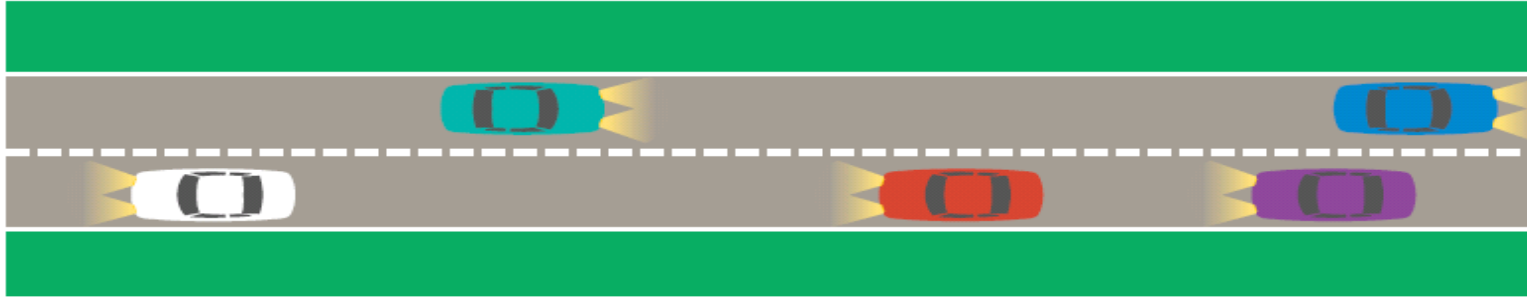


**NOTA: LE SLIDE NON RAPPRESENTANO UNA
ALTERNATIVA AL LIBRO DI TESTO
O ALLE LEZIONI DEL DOCENTE**

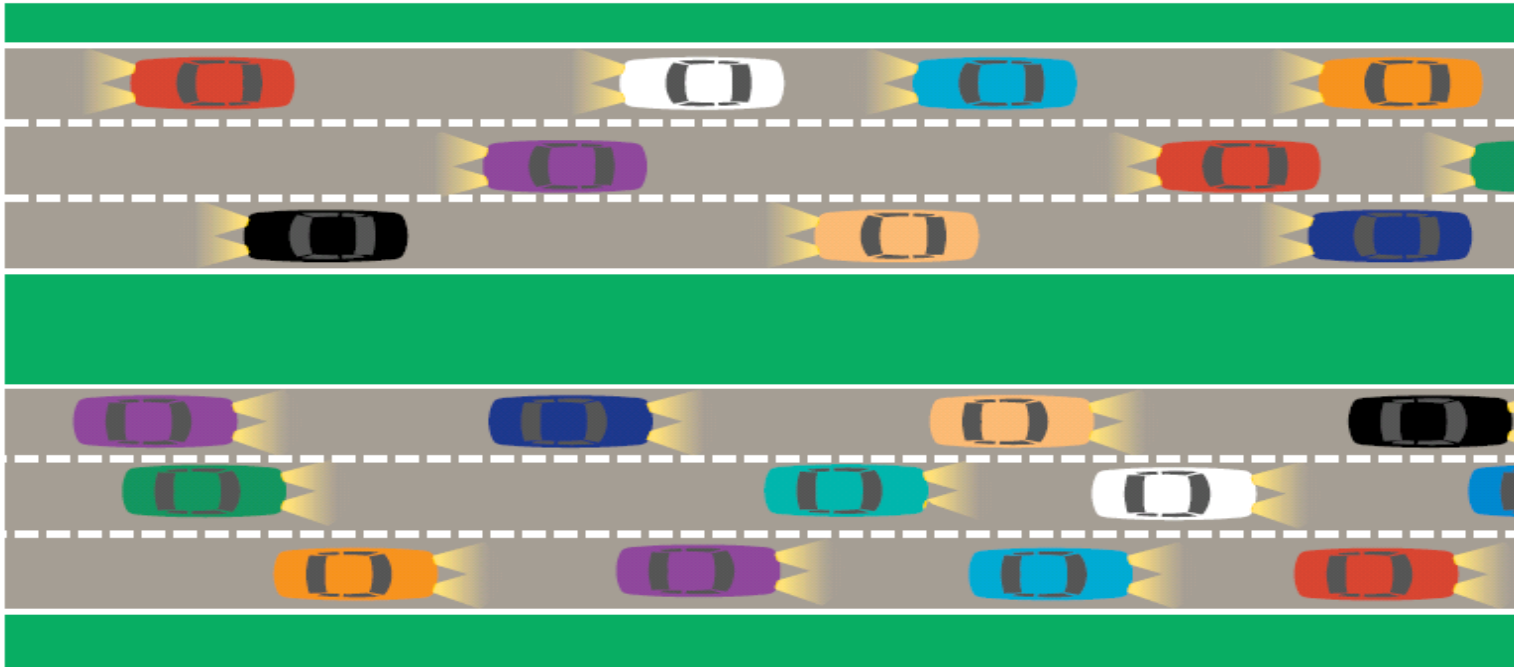
Capitolo 17

Le reti di computer:
questioni tecniche

La larghezza di banda



Una banda stretta è come una strada a due corsie



Una banda larga è come un'autostrada

La larghezza di banda

Perché la banda larga?

Banda larga significa:

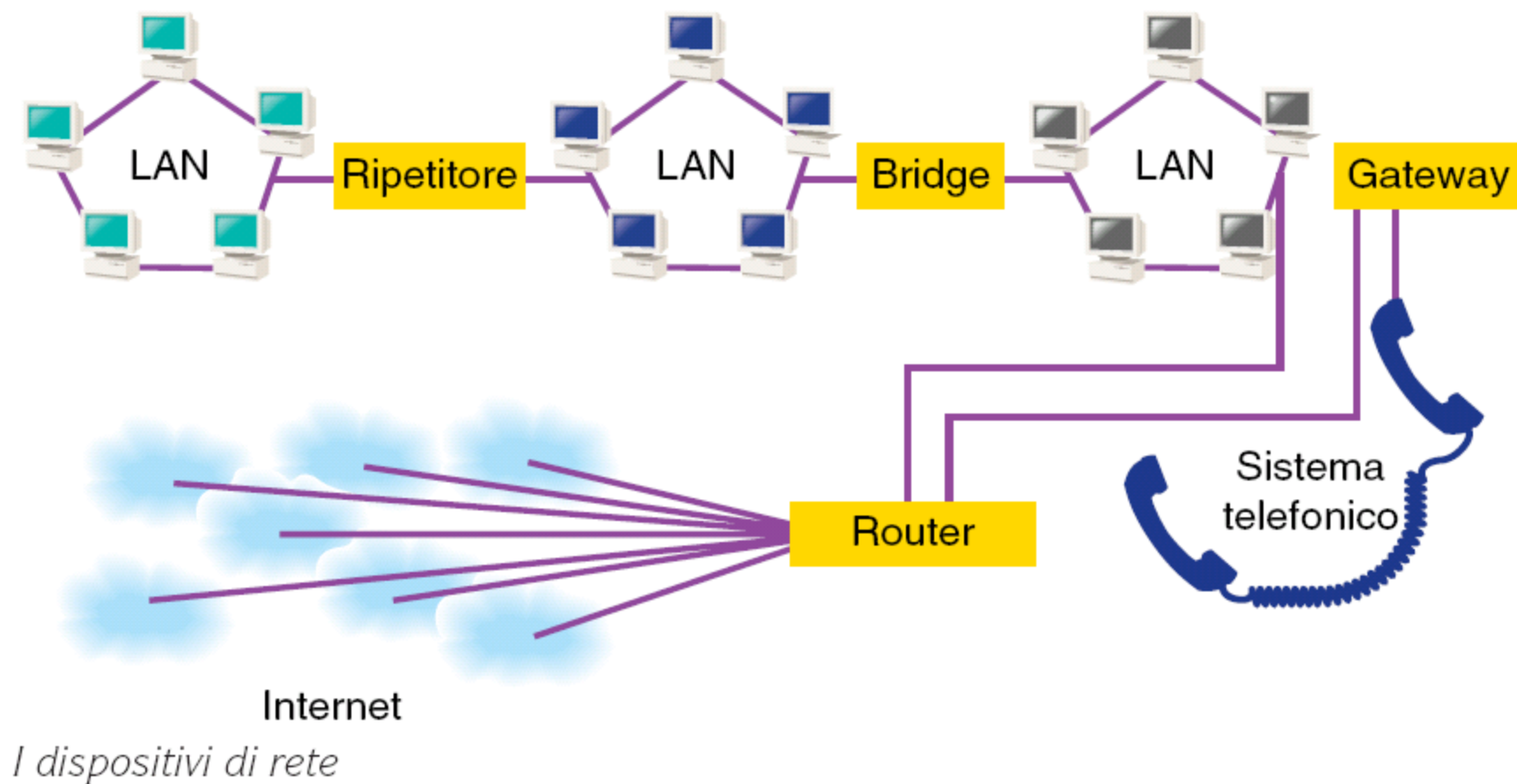
- più servizi (TV, telefonia tradizionale, “on demand”, etc.) sulla stessa linea e senza cambiare il supporto fisico;
- contenuti più “pesanti” (e interessanti).

Ecco quanti dati si devono trasmettere per un video di qualità media:

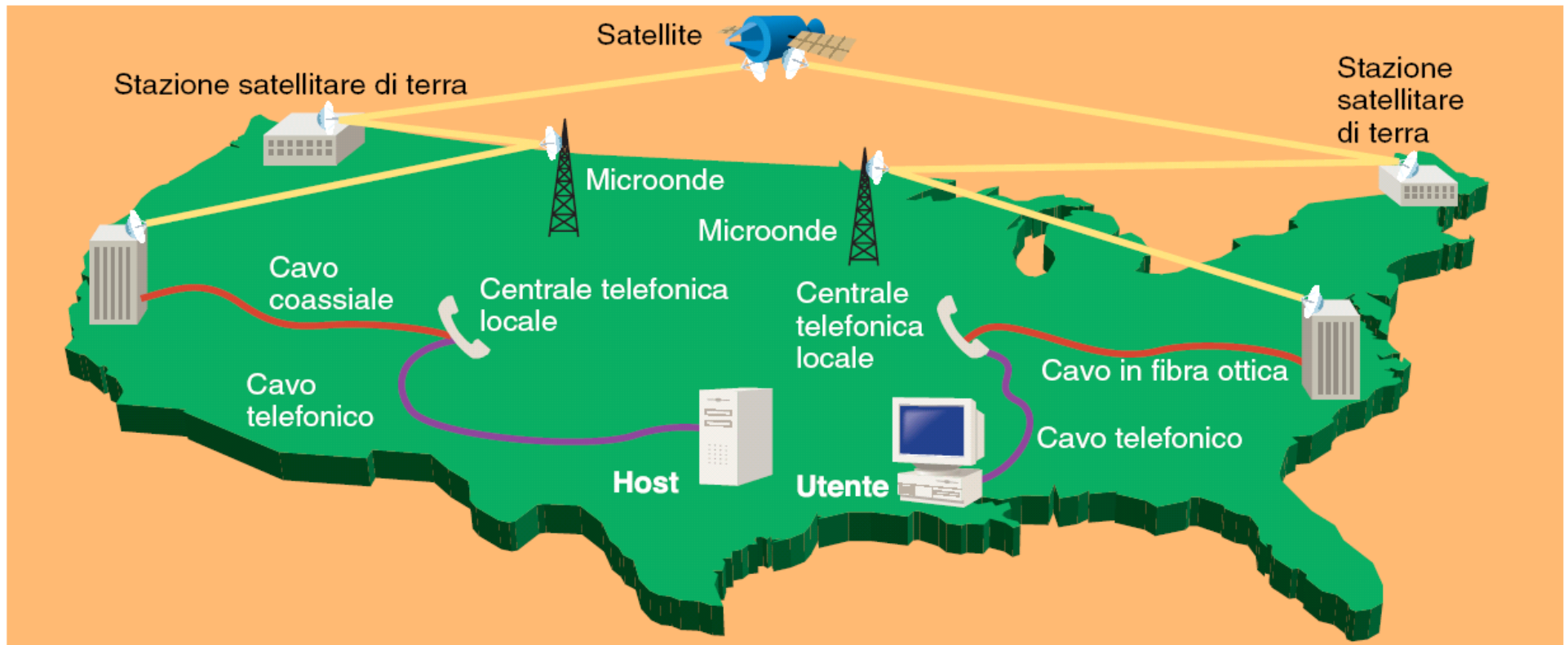
Elemento	Quantità
Pixel dello schermo (640×480)	307 200
Bit per pixel	8
Totale bit dello schermo	2 457 600
Fotogrammi per secondo	30
Totale bit per secondo	73 728 000

Le cifre della trasmissione video

Reti diverse, collegamenti diversi



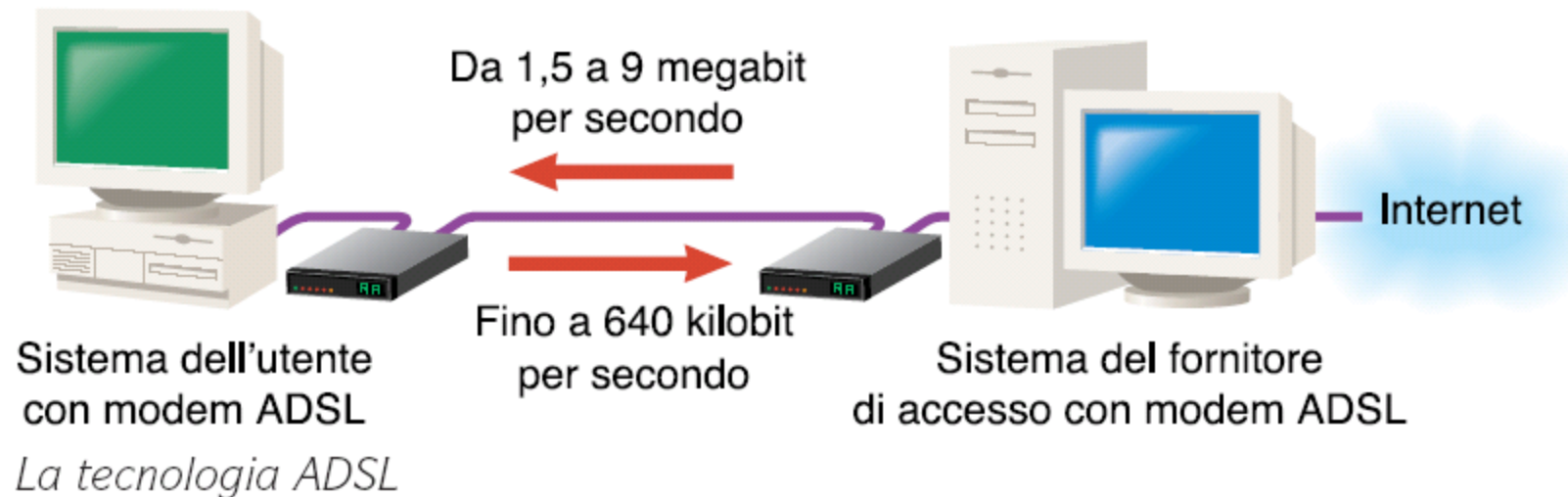
Utilità del collegamento misto



Sistema a collegamento misto

Il collegamento misto è l'unico modo per superare barriere geografiche in modo efficiente ed economico.

Vantaggi della connessione ADSL

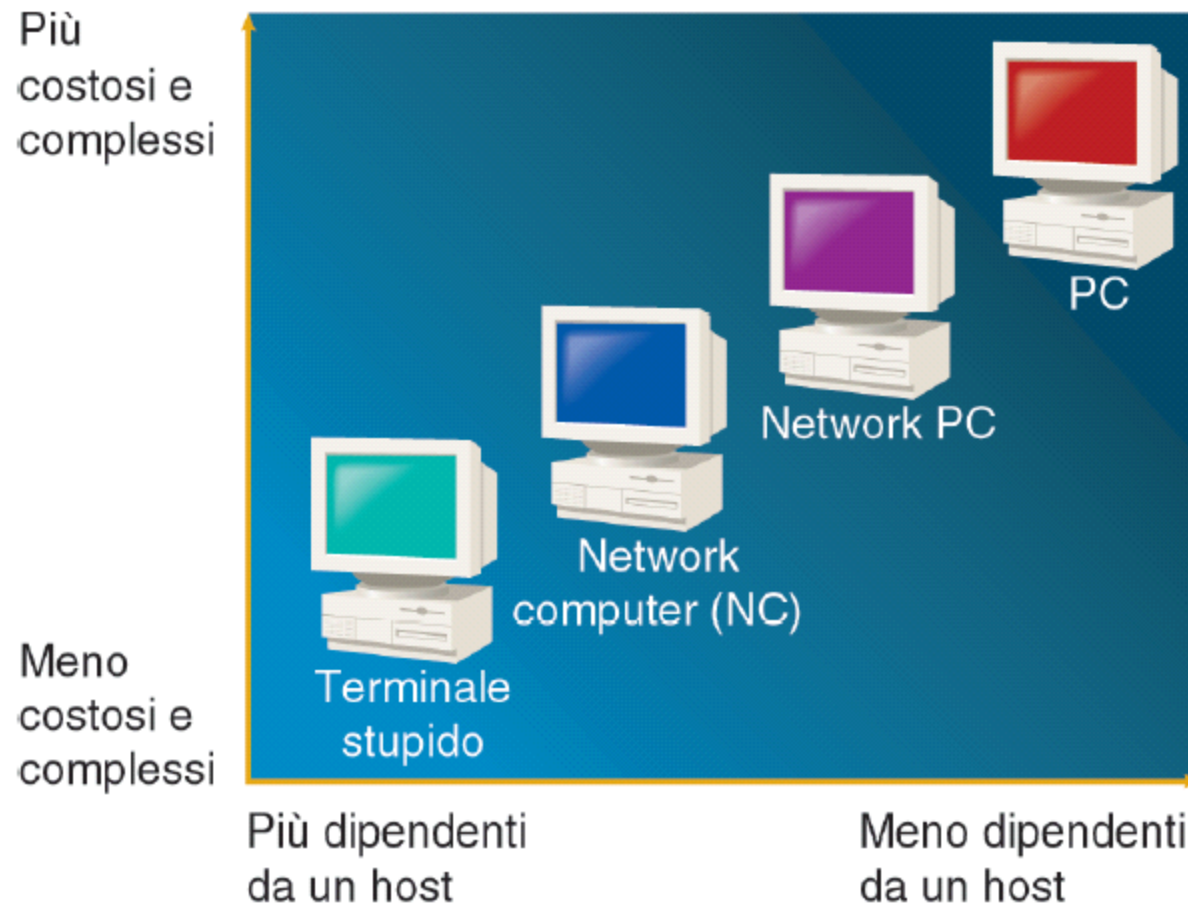


- Usa le normali connessioni in rame della telefonia.
- Resiste molto bene alle interferenze.
- La connessione alla rete è sempre attiva.

Alternative alla ADSL

- Fibra ottica:
 - più efficiente dell'ADSL, ma meno diffusa dato l'elevato costo della posa dei cavi dedicati.
- Connessioni senza fili:
 - GPRS/UMTS: permette di usare il telefono cellulare come un modem, e di scambiare i dati anche in movimento su tutta la superficie del globo. I costi, però, sono ancora alti;
 - wireless: un segnale radio viene trasmesso in un'area circoscritta. Economicamente vantaggioso, permette a molti utenti di condividere l'accesso ma le prestazioni calano notevolmente proporzionalmente al numero di utenti connesso.

Tipi di computer usabili in rete



Vari tipi di computer e loro caratteristiche

Capitolo 18

Organizzare le informazioni:
liste, query, markup

Strutture dati: le liste

- La lista è la struttura dati più semplice.
- Può essere complesso aggiungere elementi a una lista.
- Un uso efficace è quello dello “spooler” di stampa.
- Le liste non gestiscono la *meta-informazione sui dati* (ordine dei dati, tipo, ecc.).

Strutture dati: le query

Le query sono strutture che servono a interrogare le basi di dati ottenendo i risultati richiesti dall'utente. Esse mettono in relazione i campi e i record di una tabella - o più campi e record presenti in database composto da più tabelle - a dei valori di riferimento che “estraggono” dalla base di dati solo quelle informazioni che soddisfano le condizioni espresse nella query.

```
SELECT DISTINCTROW
(Customer List).(Last Name),
(Customer Charges).Date,
(Customer Charges).Amount,
(Customer Charges).Phone

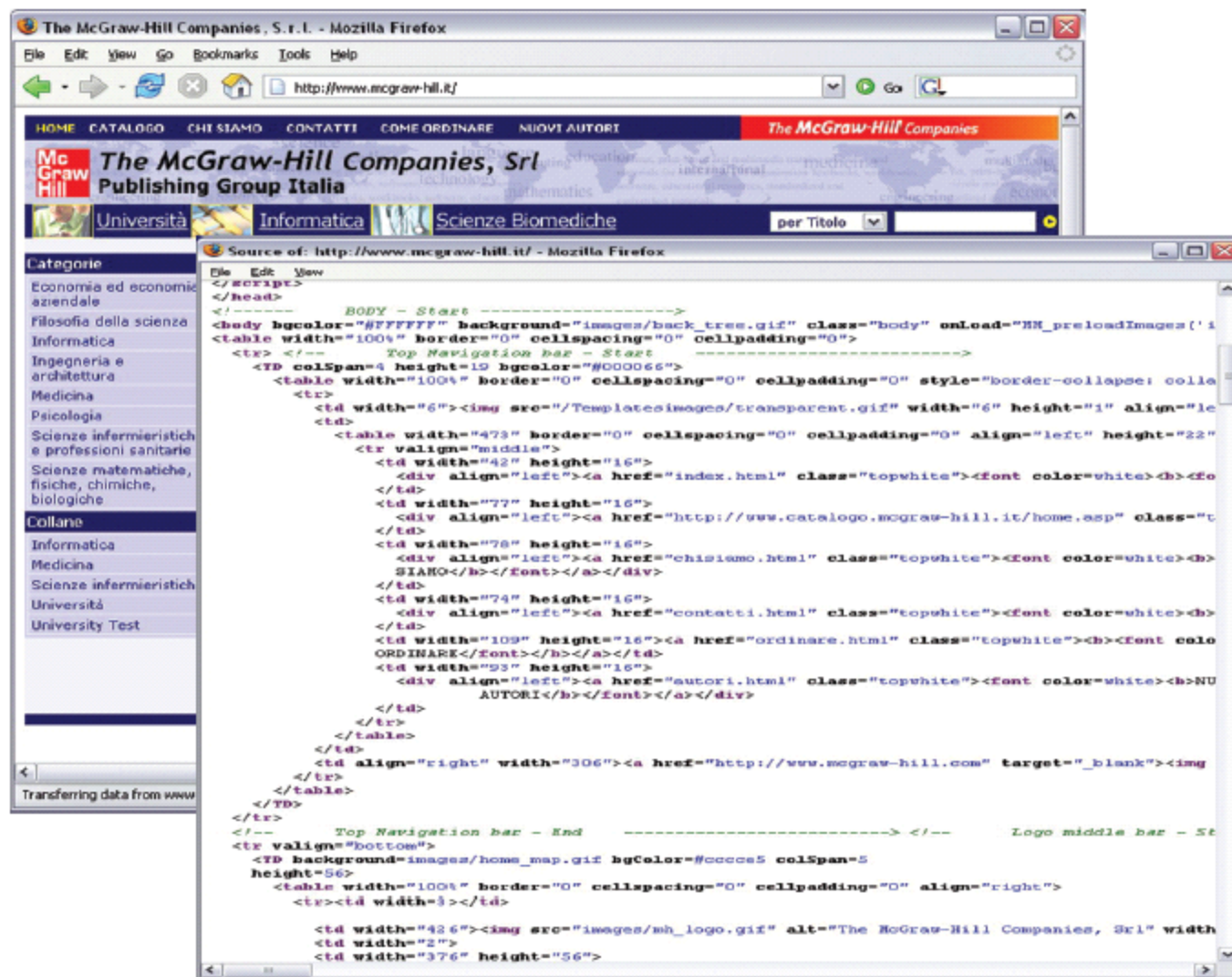
FROM
(Customer List) INNER JOIN
(Customer Charges) ON
(Customer List).ID= (Customer
Charges).ID

WHERE((((Customer List).(
Last Name))="Sen"));
```

*Una query
scritta nel
linguaggio
strutturato di
ricerca (SQL)*

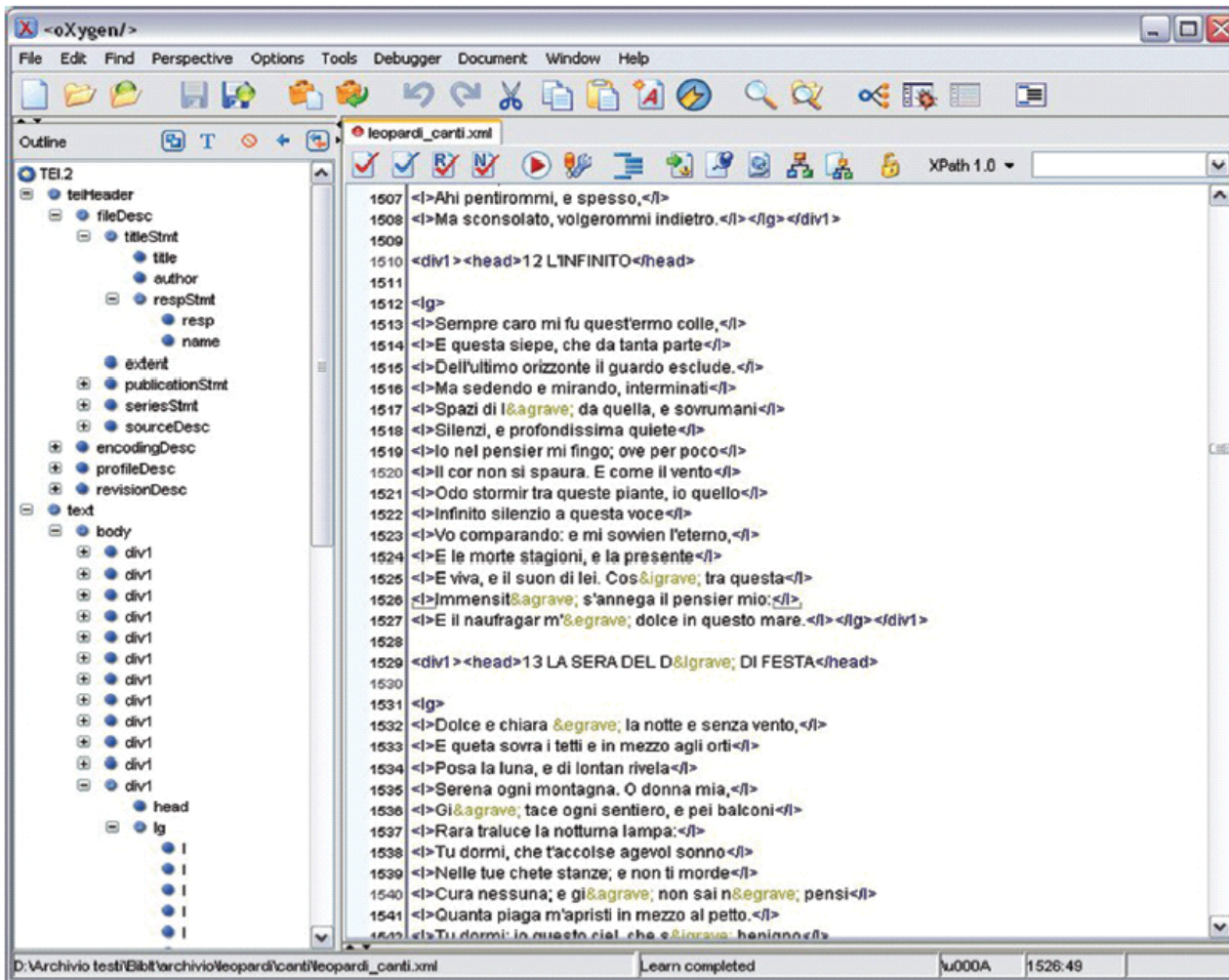
Strutture dati: HTML

Il linguaggio di markup denominato HTML è anch'esso una struttura dati, anche se la sua espressione in pagine web non sembra affatto tale. I dati da rappresentare sono diretti e contornati da tags che ne definiscono la posizione, il formato di rappresentazione, il legame con altri dati, la modifica possibile da parte dell'utente, ecc.



Un pagina Web e il relativo documento HTML

Strutture dati: XML



I Canti di Leopardi in formato XML/TEI. Si noti sulla sinistra la rappresentazione della struttura ad albero del documento XML

La struttura dati possibile con il linguaggio di markup XML è ancora più complessa di quella HTML: si può infatti aggiungere una serie di controlli definibili grazie ai quali definire gerarchicamente delle classi e dei documenti legati tra loro da vincoli semantici, e non solo sintattici. Questi vincoli possono essere efficacemente rappresentati come un “albero” i cui nodi rappresentano le relazioni esistenti tra i dati.

HTML

(HyperText Markup Language)

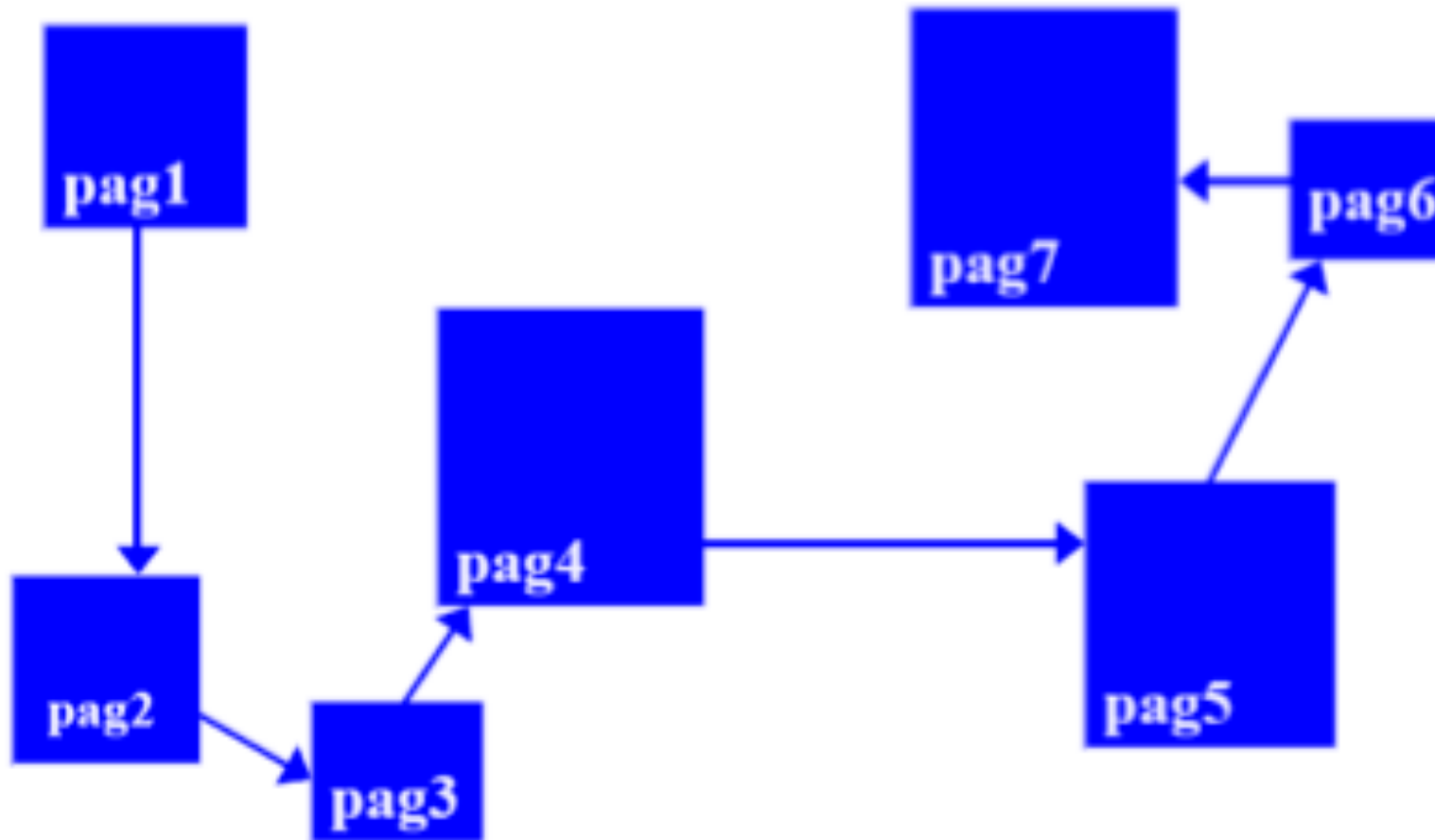


World Wide Web

- Si basa sul protocollo http ed è la vera novità degli anni '90.
- Sviluppato presso il CERN di Ginevra (da Tim Berners-Lee) è il più potente mezzo di diffusione telematica di documenti elettronici.
- Mezzo di comunicazione globale, interattivo, multimediale e ipertestuale ha cambiato radicalmente il modo di comunicare e di lavorare.

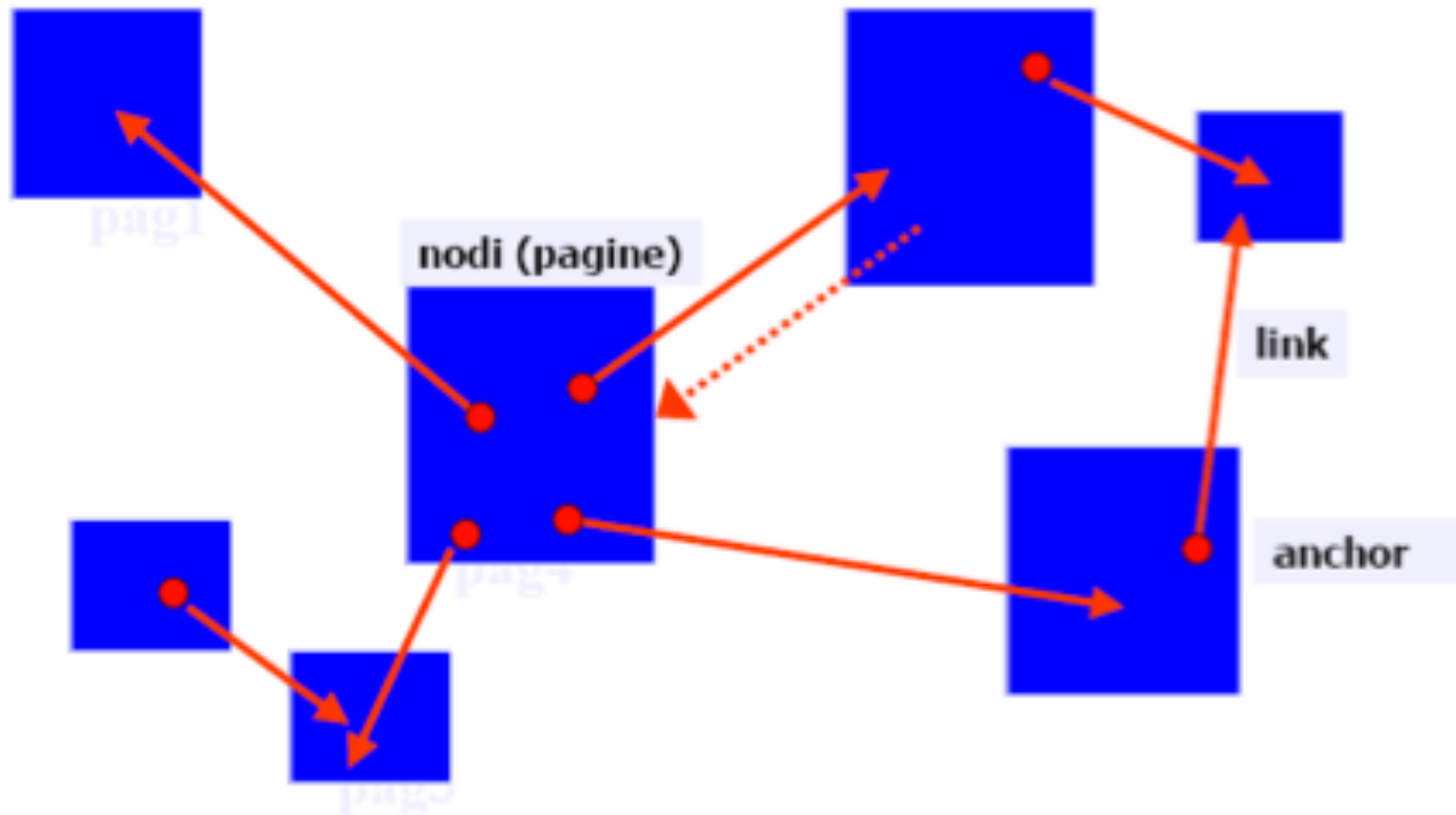
Organizzazione di un testo

Accesso sequenziale



Organizzazione di un ipertesto

Accesso secondo legami associativi



L'ipertesto è semplicemente un testo che funziona come un link.

Linguaggi di markup (1/2)

- In un documento il markup è il codice che contiene le informazioni per la sua formattazione



Linguaggi di markup (2/2)

- Nell'ambito dell'elaborazione elettronica dei testi sono stati creati strumenti informatici per automatizzare alcune operazioni per la creazione dei documenti.
- Esistono programmi per l'elaborazione dei testi di tipo diverso.
 - Gli editor **WYSIWYG**.
 - Gli editor che usano il concetto di **markup**.
- Nel caso di editor che usano il concetto di markup, un testo è costituito da due parti.
 - Il contenuto vero e proprio.
 - Le “istruzioni” che specificano come il contenuto deve essere rappresentato sul dispositivo (lo schermo di un PC, ma anche per esempio una stampante).
- In genere si racchiude il testo fra istruzioni chiamate **tag** (o etichette, o codici).
- I tag sono scritti usando i normali caratteri e quindi i documenti possono essere creati usando semplici editor di testo.

Come funziona la navigazione Internet?

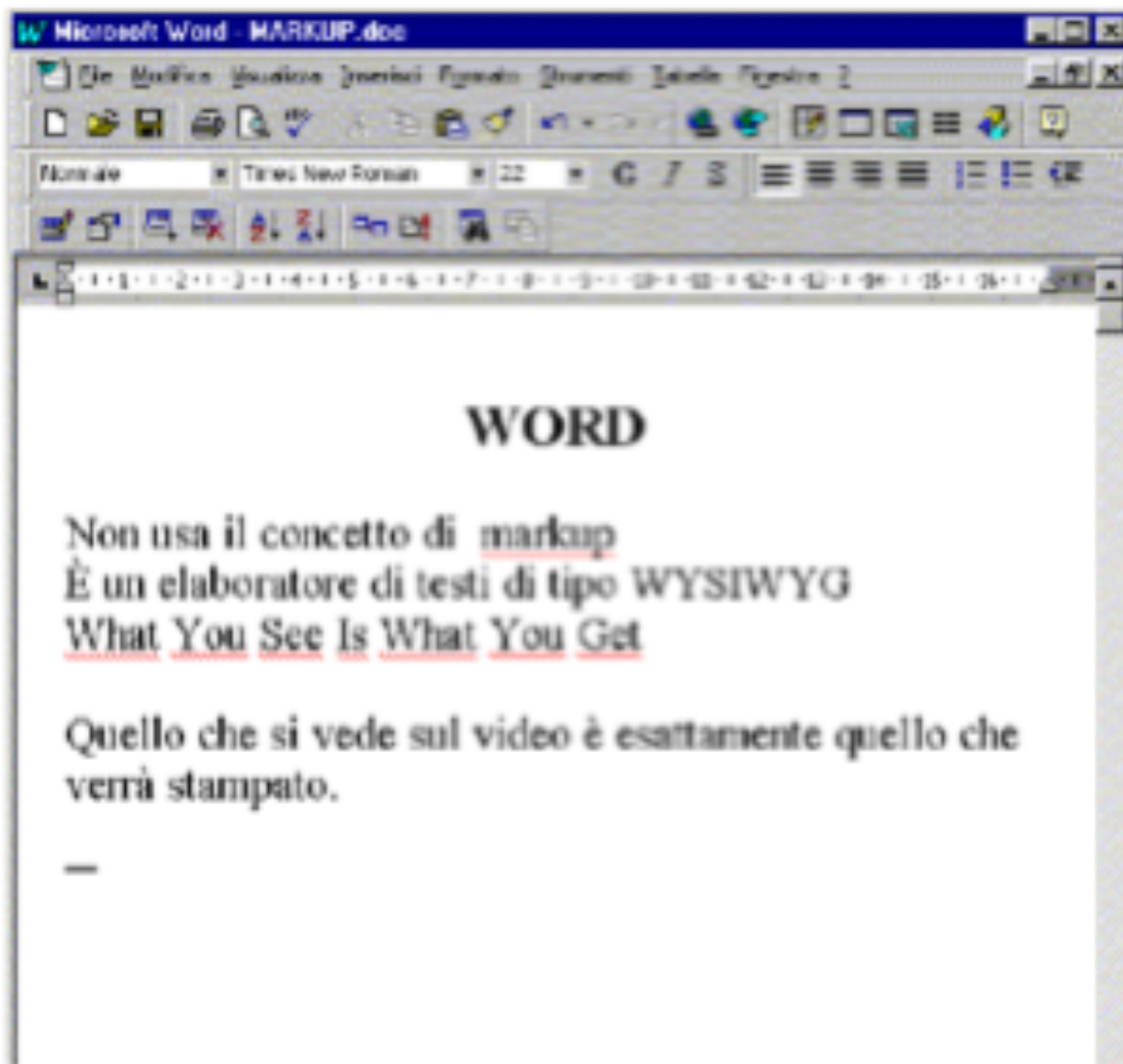


ISP



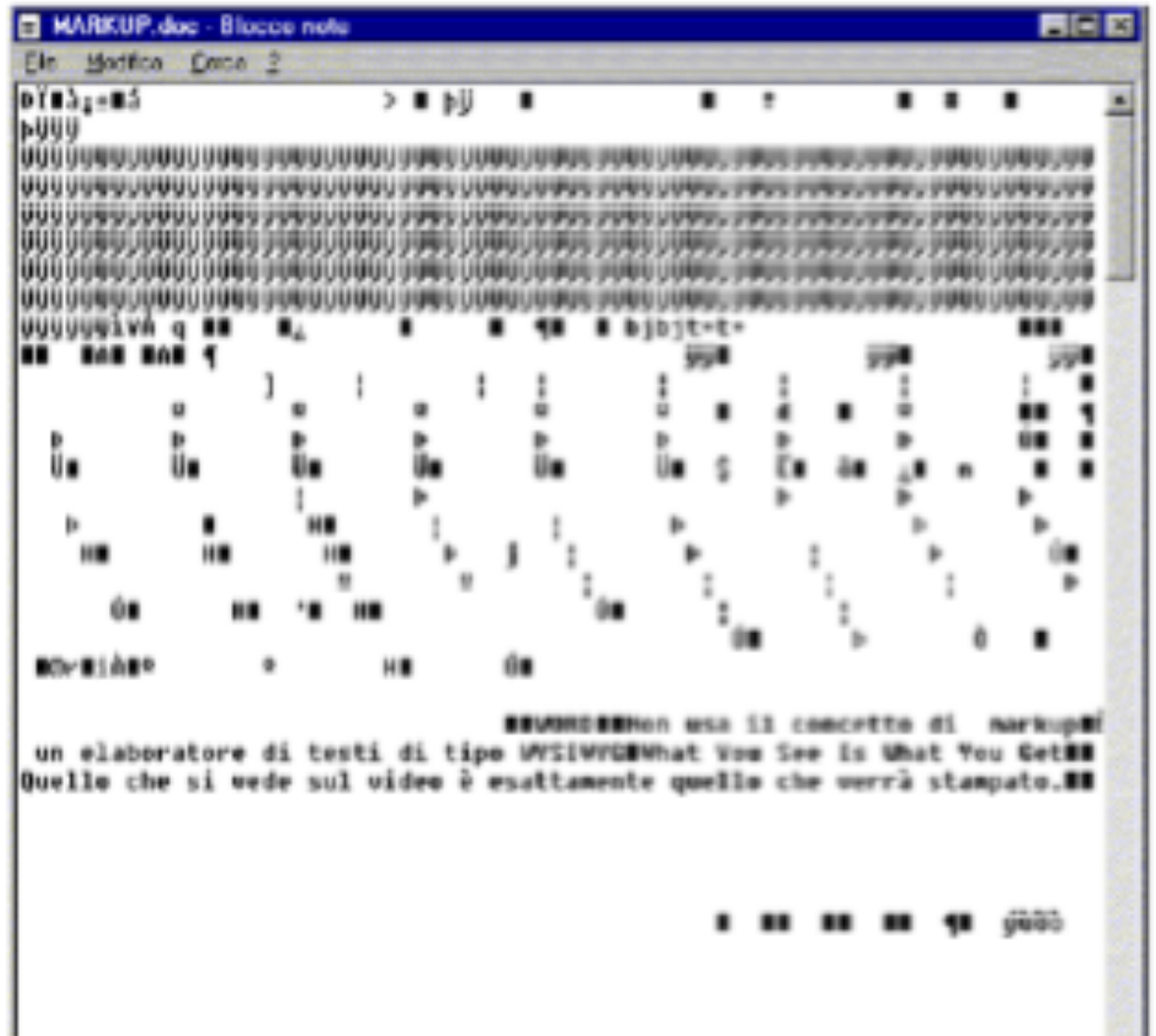
Server

File Word (.doc)



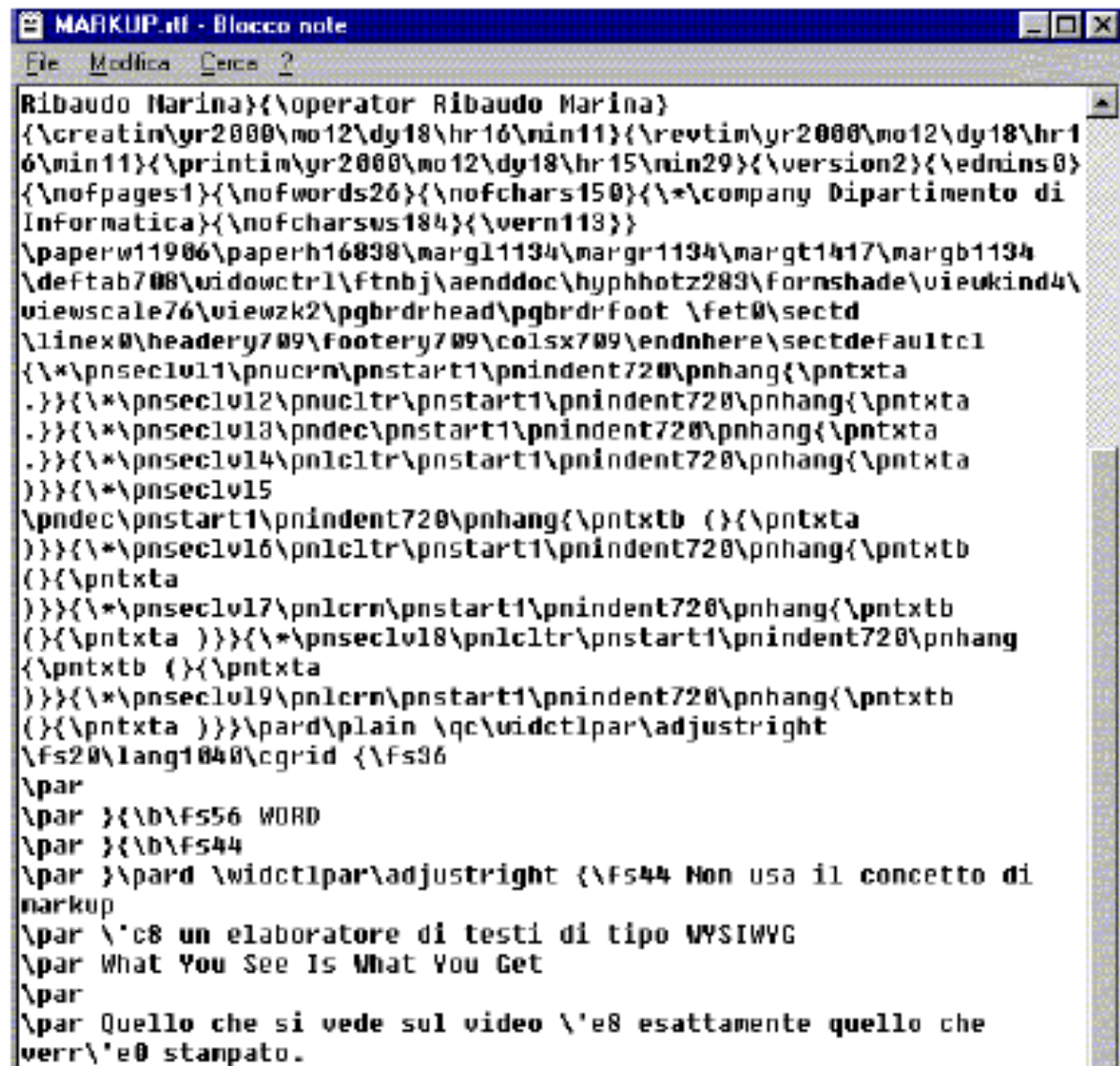
File Word (.doc)

Aprendo il file precedente con Blocco Note si vede il formato interno di Word e non si distinguono markup.



File in formato RTF (.rtf)

RTF è un formato solo testo in cui si distinguono informazioni legate alla presentazione del documento e informazioni che fanno parte del contenuto

A screenshot of a Notepad window titled 'MARKUP.rtf - Blocco note'. The window displays the raw RTF source code for a document. The code includes header information such as creator, printer, version, and company name. It also contains extensive formatting codes for paper size, margins, and font settings. The main body of the document is a series of paragraphs, some of which are indented. The text is in Italian and discusses the concept of 'markup' and the 'What You See Is What You Get' principle.

```
MARKUP.rtf - Blocco note
File Modifica Cerca ?

Ribaudo Marina}{\operator Ribaudo Marina}
{\creatin\yr2000\mo12\dy18\hr16\min11}{\revtim\yr2000\mo12\dy18\hr16\min11}{\printim\yr2000\mo12\dy18\hr15\min29}{\version2}{\edmins0}
{\nofpages1}{\nofwords26}{\nofchars150}{\*\company Dipartimento di Informatica}{\nofcharsus184}{\vern113}}
\paperw11906\paperh16838\margl1134\margr1134\margt1417\margb1134
\defab708\widowctr1\ftnbj\aeenddoc\hyphhotz283\formshade\viewkind4\
viewscale76\viewwk2\pgbrdrhead\pgbrdrfoot \fet0\sectd
\linex0\headery709\footery709\colsx709\endnhere\sectdefaultc1
{\*\pnseclvl1\pnucrn\pnstart1\pnindent720\pnhang{\pntxta
.}}{\*\pnseclvl2\pnucltr\pnstart1\pnindent720\pnhang{\pntxta
.}}{\*\pnseclvl3\pndec\pnstart1\pnindent720\pnhang{\pntxta
.}}{\*\pnseclvl4\pnlcltr\pnstart1\pnindent720\pnhang{\pntxta
.}}{\*\pnseclvl5
\pndec\pnstart1\pnindent720\pnhang{\pntxtb (}}{\pntxta
)}}{\*\pnseclvl6\pnlcltr\pnstart1\pnindent720\pnhang{\pntxtb
(}}{\pntxta
)}}{\*\pnseclvl7\pnlcrn\pnstart1\pnindent720\pnhang{\pntxtb
(}}{\pntxta )}}{\*\pnseclvl8\pnlcltr\pnstart1\pnindent720\pnhang
{\pntxtb (}}{\pntxta
)}}{\*\pnseclvl9\pnlcrn\pnstart1\pnindent720\pnhang{\pntxtb
(}}{\pntxta )}}\pard\plain \qc\widctlpar\adjustright
\fs20\lang1040\cgrid {\fs36
\par
\par }{\b\fs56 WORD
\par }{\b\fs44
\par }\pard \widctlpar\adjustright {\fs44 Non usa il concetto di
markup
\par \c8 un elaboratore di testi di tipo WYSIWYG
\par What You See Is What You Get
\par
\par Quello che si vede sul video \e8 esattamente quello che
verr\ e0 stampato.
```

File in formato PostScript (.ps)

Questo è un formato usato per la stampa. Anche in questo caso si distinguono informazioni legate alla presentazione del documento e informazioni che fanno parte del documento stesso.



```
markup.gen - Blocco note
File Modifica Formati
%%-12345XHPJL JOBHPJL SET RESOLUTION = 600HPJL ENTER
LANGUAGE=POSTSCRIPT
%%PS-Adobe-3.0
%%Title: Microsoft Word - MARKUP.rtf
%%Creator: PSCRIPT.DRW versione 4.0
%%CreationDate: 12/18/00 16:12:57
%%BoundingBox: 14 13 581 829
%%Pages: (atend)
%%PageOrder: Special
%%Requirements:
%%DocumentNeededFonts: (atend)
%%DocumentSuppliedFonts: (atend)
%%DocumentData: Clean7Bit
%%LanguageLevel: 2
%%EndComments

%%BeginProlog
%%BeginProcSet: Pscript_Win_ErrorHandler 1.0 1
/currentpacking where{pop/oldpack currentpacking def/setpacking
where{pop
false setpacking}if}{if/$hrkpage 64 dict def $hrkpage begin/print{dup
type
/stringtype ne{=string cvs}if dup length 6 mul/tx exch def/ty 10
def
currentpoint/toy exch def/tox exch def 1 setgray newpath tox toy 2
sub moveto
0 ty rlineto tx 0 rlineto 0 ty neg rlineto closepath fill tox toy
moveto 0
setgray show}bind def/nl{currentpoint exch pop lmargin exch moveto
0 -10
rmoveto}def/=={/cp 0 def typeprint nl}def/typeprint{dup type
end
```

HTML

- Gli ipertesti del web sono scritti usando il linguaggio **HyperText Markup Language** (HTML).
- HTML **non** è un linguaggio di programmazione!
 - HTML non ha meccanismi per prendere delle decisioni o compiere delle iterazioni.
- HTML è un **linguaggio di markup** e permette di descrivere la disposizione di tutti gli elementi presenti all'interno di un documento.
 - Le indicazioni vengono date attraverso degli appositi marcatori, detti "tag".
- I documenti HTML sono dei file in **formato testo** (codice ASCII).
- Si possono creare con degli editor di testo dando l'estensione **.htm** o **.html**.
- I browser leggono i documenti HTML e li visualizzano **interpretando** le specifiche di formattazione in modo sequenziale.

Standard (X)HTML

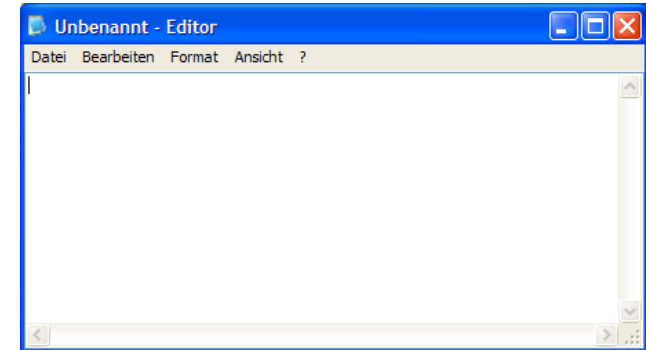
L'organizzazione che si occupa di standardizzare la sintassi del linguaggio HTML (il W3C: World Wide Web Consortium) ha rilasciato diverse versioni di questo linguaggio (HTML 2.0, HTML 3.2, HTML 4.0); e - da un certo punto in poi - l'HTML si è evoluto in XHTML (si tratta dell'HTML riformulato come linguaggio XML - ne sono già state rilasciate due versioni).

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML
2 <html>
3     <head>
4         <title>Example</title>
5         <link href="screen.css" rel="sty
6     </head>
7     <body>
8         <h1>
9             <a href="/">Header</a>
10        </h1>
11        <ul id="nav">
12            <li>
13                <a href="one/">One</a>
14            </li>
15            <li>
16                <a href="two/">Two</a>
17            </li>
```

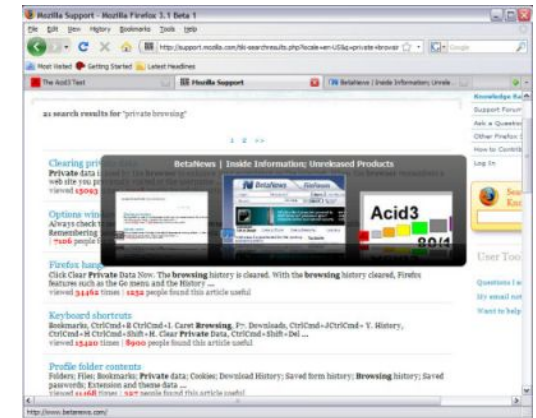


Di cosa abbiamo bisogno

✓ Un editor di testo semplice (Notepad – Blocco note – Wordpad).



✓ Un browser internet (Firefox, Explorer, Safari).



✓ Uno spazio su un server per rendere pubbliche le nostre pagine (opzionale).



HTML: un esempio

```
<html>
```

```
  <head>
```

```
    <title>Pagina HTML di prova</title>
```

```
  </head>
```

```
  <body>
```

Utilizzare un editor di testo: in ambiente Windows WordPad va bene, in ambiente Unix vi per esempio

```
  </body>
```

```
</html>
```

HTML: sintassi (1/2)

<nomecomando>

informazioni

</nomecomando>

- Un documento HTML inizia sempre con il tag <html> e termina sempre con il tag </html>.
- Il linguaggio HTML è **case-insensitive** e quindi <HTML>, <html>, <Html> sono tutti tag leciti (lo stesso vale per gli altri tag).

Suggerimenti:

- e meglio decidere all'inizio come si scriveranno i tag ed essere consistenti;
- per analogia con XHTML (derivato dell'XML e dell'HTML che è case-sensitive) è consigliabile scrivere tutto in minuscolo.

HTML: sintassi (2/2)

`<html>`

`<head>`

caratteristiche del documento

`</head>`

`<body>`

documento

`</body>`

`</html>`

HTML: <head>

<html>

<head>

<title>titolo del documento</title>

<meta name="keywords" content="parole chiave qui">

<meta name="author" content="nome e cognome qui">

</head>

.....

.....

</html>

HTML: <head>

- Questa sezione del documento contiene informazioni non immediatamente percepibili, ma che riguardano il modo in cui il documento deve essere letto ed interpretato.
- È la parte del documento dove vengono inseriti i metatag (alcuni sono ad esclusivo beneficio dei motori di ricerca), script JavaScript o VbScript, i fogli di stile, ecc.

HTML: <body>

<html>

<head>

<title>titolo del documento</title>

</head>

<body lista di opzioni>

documento vero e proprio

</body>

</html>

HTML: <body>

<body

bgcolor="colore sfondo"

background="nome file con l'immagine"

text="colore testo"

link="colore link da visitare"

vlink="colore link visitato"

alink="colore link selezionato"

>

NB: l'ordine in cui si scrivono gli attributi non è importante.

HTML: <body>

bgcolor="red"

bgcolor="yellow"

bgcolor="#00ffff"

text="#eeeeee"

link="#ffffff"

background="images/sfondo.gif"

<body text= "white" link="#cc0000" background="images/sabbia.gif">

HTML: RGB (1/2)

- Ogni colore può essere codificato mediante tre numeri compresi tra 0 e 255 che rappresentano le quantità di **ROSSO** **VERDE** e **BLU** presenti nel colore stesso.
- Ogni numero compreso tra **0** e **255** deve essere trasformato nella rappresentazione **esadecimale** corrispondente (base 16).
- Con questa codifica si possono rappresentare più di 16,7 milioni di colori diversi.

HTML: RGB (2/2)

	decimale	#esadecimale
nero	0 0 0	#000000
blu	0 0 255	#0000ff
verde	0 255 0	#00ff00
rosso	255 0 0	#ff0000
 bianco	255 255 255	#ffffff

NB: nei file HTML le codifiche esadecimali vanno scritte senza lasciare spazi vuoti tra le cifre.

HTML: titoli

- Il documento inserito nel `<body>` viene visualizzato secondo le direttive di formattazione.

Per scrivere **titoli** si
possono usare:

<code><h1></code>	<code></h1></code>
<code><h2></code>	<code></h2></code>
<code><h6></code>	<code></h6></code>

Per andare **a capo**:

<code>
</code>
<code><p>.....</p></code>

HTML: caratteri

Formato dei caratteri:

- **** grassetto ****
- **** grassetto ****
- **<i>** corsivo **</i>**
- **** enfatizzato ****
- **<code>** codice **</code>**
- **** testo ****
- **** ... ****
- **** ... ****

HTML: allineamento

- Allineamento:
 - `<p align="center">`
 - `<p align="right">`
 - `<p align="justify">`
 - `<center>` testo da centrare `</center>`

HTML: elenchi

- **** e **** (lista non ordinata)

****primo elemento****

****secondo elemento****

- **** e **** (lista ordinata)

****primo elemento****

****secondo elemento****

HTML: altri marcatori

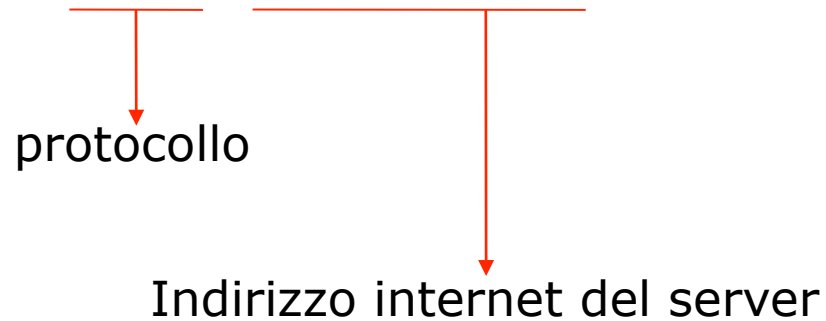
- Linee orizzontali `<hr>`:
 - `<hr width="90%" size=3>`
- Testo lampeggiante (solo per Netscape):
 - `<blink>` questo testo lampeggia `</blink>`
- Testo scorrevole (solo per Explorer):
 - `<marquee>` questo testo scorre `</marquee>`
- Commenti:
 - `<!--` questo testo non viene visualizzato `-->`

HTML: <a href>

- Link <a>
 - testo del link
- Gli indirizzi per “trovare” i documenti nel web sono detti **Uniform Resource Locator (URL)** e li identificano in modo univoco.
- Protocollo: //indirizzo internet del server/pathname

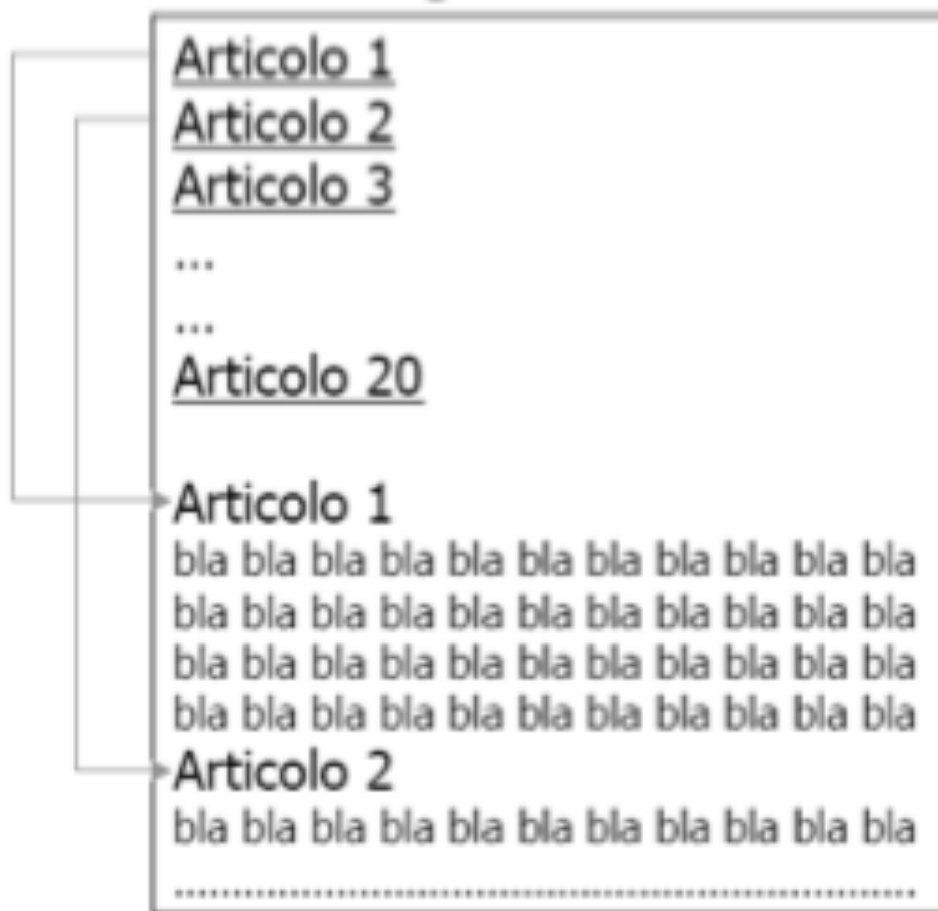
HTML: <a href>

Università di Bari



HTML: link interni (ancore) (1/3)

- Link interni allo stesso documento: si usa l'attributo **name** del tag **<a>**



HTML: link interni (ancore) (2/3)

- Si devono marcare quelle porzioni del documento dove si vuole saltare.

```
Articolo 1  
Articolo 2  
Articolo 3  
...  
...  
Articolo 20  
  
Articolo 1 <a name="art1"></a>  
bla bla bla bla bla bla bla bla bla bla  
bla bla bla bla bla bla bla bla bla bla  
bla bla bla bla bla bla bla bla bla bla  
bla bla bla bla bla bla bla bla bla bla  
Articolo 2 <a name="art2"></a>  
bla bla bla bla bla bla bla bla bla bla  
.....
```

HTML: link interni (ancore) (3/3)

- Per i link interni si deve usare nell'attributo href il nome scelto come valore dell'attributo **name**, preceduto dal simbolo #.

```
Articolo 1 <a href="#art1">Articolo1 </a>
Articolo 2 <a href="#art2">Articolo2 </a>
Articolo 3 <a href="#art3">Articolo3 </a>
...
...
Articolo 20

Articolo 1 <a name="art1"></a>
bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla
Articolo 2 <a name="art2"></a>
bla bla bla bla bla bla bla bla bla bla
*****
```

HTML:

NB: l'immagine deve essere su un file a parte in formato GIF, PNG o JPEG (BMP solo per Explorer).

HTML: audio

- Formati audio .au .aiff .mid .wav
 - `<a href="nome file audio">clicca qui</a>`
 - `<bgsound src="nome file audio" loop="10">`

NB: bgsound fa partire l'audio quando si entra nella pagina ma funziona solo in Explorer.

HTML: video

- Formati video .mpeg .avi .mov
 - `<a href="nome file video">clicca qui</a>`
 - ``

NB: dynsrc fa partire automaticamente il video ma funziona solo in Explorer.

HTML: tabelle

- Per creare una tabella in HTML si usano i tag:

<table> **</table>**

all'interno dei quali si deve definire il contenuto della tabella

- Per definire una tabella si procede nel modo seguente:
 - Si realizza la prima riga facendo uso del tag **<tr>** (table row)
 - Si indicano le varie celle facendo uso:
 - Del tag **<th>** per le intestazioni di cella **</th>** (table header)
 - Del tag **<td>** per i dati della cella **</td>** (table data)
 - Si passa alla seconda riga con le relative celle avendo cura di usare il tag **</tr>**

HTML: tabelle

HTML: <table>

<table

width="larghezza della tabella"

cellpadding="numero"

indica la distanza tra il contenuto della cella e il bordo. Se il valore viene indicato con un numero intero, la distanza è espressa in pixel; il cellpadding tuttavia può anche essere espresso in percentuale. Di default la distanza è nulla.

cellspacing="numero"

specifica la distanza (in pixel) tra una cella e l'altra, oppure tra una cella e il bordo. Di default è un pixel, dunque occorrerà sempre azzerarlo esplicitamente, quando non lo si desidera.

border="numero"

specifica la larghezza dei bordi di una tabella (in pixel).

bgcolor="colore sfondo tabella"

Esempio:

<table **width**="80%" **cellpadding**="5" **bgcolor**="#ff3300">

HTML: <td> <th>

<td (th)

width="dimensione della cella"

colspan="numero"

rowspan="numero"

bgcolor="colore sfondo cella"

align="right" | "left" | "center"

valign="top" | "bottom" | "middle"

nowrap

>

Esempio:

- `<td width="100" bgcolor="#0033aa" align="right">`

HTML: tabelle

- Le tabelle si usano principalmente per organizzare gli elementi all'interno di una pagina.
- Per fare questo non si visualizzano i bordi

`<table border="0">`

righe e colonne

`<\table>`

XML

(eXtensible Markup Language)



Introduzione

La tecnologia che va sotto il nome di
“ eXtensible Markup Language (XML)”
è una delle più complesse e controverse
che siano entrate in scena negli ultimi anni.

World Wide Web Consortium (W3C)

Il W3C è un consorzio fondato nell'ottobre 1994 con l'intento di “condurre il World Wide Web alla sua piena funzionalità” attraverso lo sviluppo di protocolli comuni che promuovano la sua evoluzione ed assicurino l'interoperabilità, ossia lo scambio di informazione tra qualsiasi tipo di utente con qualsiasi mezzo.

Le origini

Il linguaggio XML ha origine da SGML (Standard Generalized Markup Language), un meta-linguaggio “markup” nato per sviluppare informazioni strutturate e che, di fatto, offre agli utenti la possibilità di creare descrizioni molto complesse di documenti e dati.

Da SGML a XML

- SGML è potente e flessibile ma la sua complessità ha spinto il World Wide Web Consortium (W3C) a ridefinire una sua parte, quindi un nuovo meta-linguaggio, che va sotto il nome di XML.
- XML doveva rilevare parte della potenza di SGML e renderla più lineare e semplice da utilizzare.

Da SGML a XML

- In breve tempo sono state sviluppate altre tecnologie di supporto che consentono l'elaborazione dei dati rappresentati da XML tramite strumenti standard.
 - XSL, XSLT, Xpath, XPointer, Xquery, etc.
- XML mette a disposizione una serie di regole per poter definire un nostro linguaggio per la rappresentazione di informazioni.
- Qualsiasi contesto può avvalersi di questa tecnologia:
 - rappresentazione di database, interscambio di dati, generazione di pagine web, ecc.

XML

Il linguaggio XML descrive:

- il modo in cui scrivere documenti XML
 - file di testo contenenti dati strutturati;
- il modo in cui tali file devono essere preliminarmente processati affinché un'applicazione possa poi accedere ai dati in essa contenuti.

Obiettivi di XML

Gli obiettivi dello standard XML sono:

- avere un linguaggio indipendente dalla piattaforma hardware utilizzata;
- utilizzabile in Internet;
- compatibile con il linguaggio standard internazionale SGML ,utilizzato per la rappresentazione di documenti e dati strutturati.

Struttura dei documenti XML

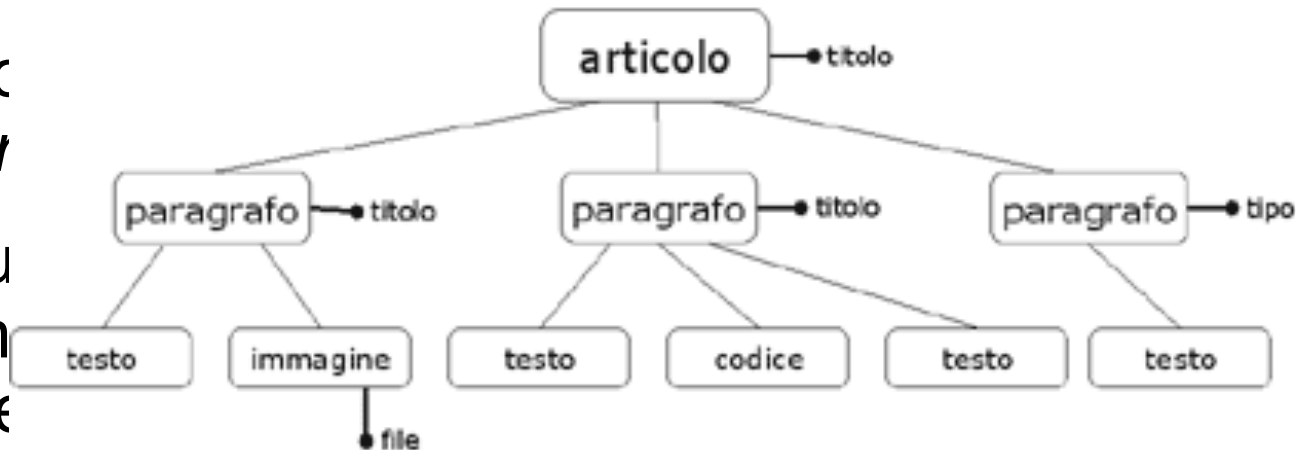
Un documento XML deve essere visto come l'unione di due strutture:

- la struttura logica;
- la struttura fisica.

Struttura logica (1/2)

- Un documento è composto da elementi

- Ciascun elemento ha un tipo e un titolo



inatti

gico del
ementi)

- Agli elementi si possono associare altre informazioni che ne descrivono le proprietà (attributi).

Struttura logica (2/2)

- L'organizzazione di questi elementi segue un ordine gerarchico:
 - esiste un elemento principale, detto *root element* (o *root*), che contiene l'insieme degli altri elementi del documento.
- Questa struttura gerarchica può essere rappresentata graficamente tramite un albero, detto *document tree*.

Struttura fisica

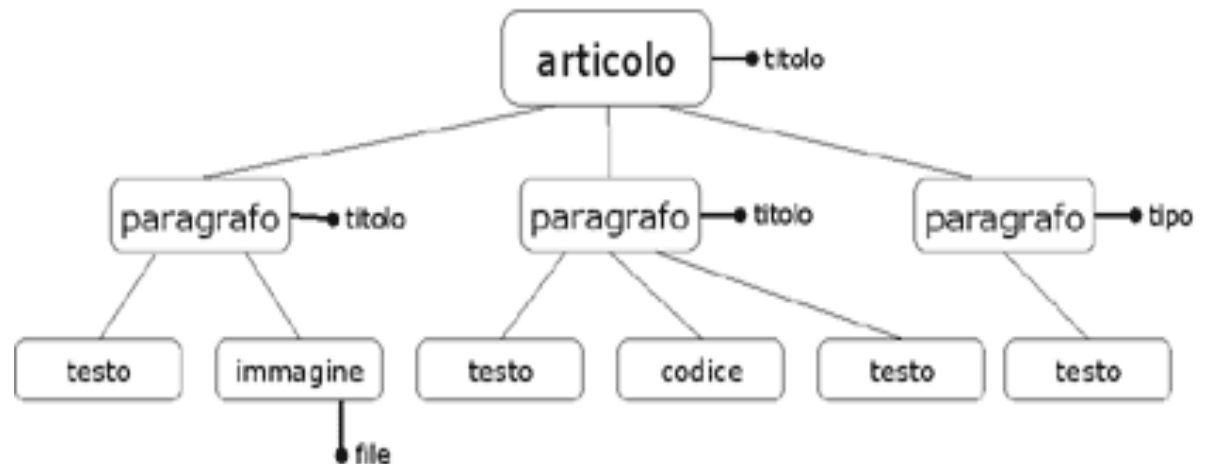
- Un documento XML è un file di testo che può essere creato con un qualsiasi editor
 - AMAYA, VISUAL XML, XED, XML Notepad, XML<Pro>
- Ogni documento XML deve essere registrato in un file con estensione **.xml**
- I documenti XML sono nel formato testo, in modo che siano facili da:
 - scrivere;
 - leggere;
 - correggere;
 - scambiare tra utenti che utilizzano sistemi software diversi tra loro.

Esempio di struttura fisica

```
<?xml version="1.0" ?>
<articolo titolo="Titolo dell'articolo">
  <paragrafo titolo="Titolo del primo paragrafo">
    <testo>
      Blocco di testo del primo paragrafo
    </testo>
    <immagine file="immagine1.jpg">
    </immagine>
  </paragrafo>
  <paragrafo titolo="Titolo del secondo paragrafo">
    <testo>
      Blocco di testo del secondo paragrafo
    </testo>

    <codice>
      Esempio di codice
    </codice>
    <testo>
      Altro blocco di testo
    </testo>
  </paragrafo>

  <paragrafo tipo="bibliografia">
    <testo>
      Riferimento ad un articolo
    </testo>
  </paragrafo>
</articolo>
```



Struttura del documento

Ogni documento XML è costituito da un prologo e da una document entity.

- Il *prologo* è formato da istruzioni e dichiarazioni che vengono utilizzate per l'interpretazione del documento XML.
- La *document entity* è il corpo centrale del documento XML ed è formata da elementi, attributi ed entità.

Prologo

Il prologo è costituito da:

1. dichiarazione XML;
2. dichiarazione del tipo di documento;
3. dichiarazione di entità.

Dichiarazione XML

- La dichiarazione XML deve essere sempre presente.
- Ha la forma

`<?xml version="1.0"?>`

- Definisce il documento come documento XML.
- Specifica la versione del linguaggio XML utilizzata.
- Specifica se si tratta di un documento standalone, ossia se non necessita di una DTD (Document Type Definition).

Esempio di dichiarazione XML

La dichiarazione

```
<?xml version="1.0" standalone="no" ?>
```

indica che:

- il documento XML utilizza il set di regole della versione 1.0 dello standard XML;
- il documento non è standalone e quindi necessita di una DTD esterna.

Dichiarazione del tipo di documento

La dichiarazione del tipo di documento:

- è opzionale;
- fornisce la specifica o il collegamento ad una grammatica (DTD);
- deve sempre seguire la dichiarazione XML.

Dualità del documento

Un documento XML può essere descritto da un punto di vista fisico o da un punto di vista logico:

- **fisicamente** un documento XML è formato da entità;
- **logicamente** è formato da elementi, attributi e contenuto.

Dichiarazione di entità

La dichiarazione di entità:

- è opzionale;
- serve per dichiarare un'entità che verrà utilizzata all'interno del documento;
- deve seguire la dichiarazione del tipo di documento.

Entità

- Un'entità è un insieme di dati che si ripete sempre uguale in più punti della struttura del documento.
- Ogni entità è identificata da:
 - un nome univoco;
 - un valore, che può essere costituito da:
 - una stringa di caratteri;
 - un documento XML;
 - un file esterno di qualsiasi formato (immagine, suono, video o altro).

Dichiarazione di entità

- Le entità vengono dichiarate nel prologo mediante una dichiarazione del tipo:

<!ENTITY ... >

- Un'entità può essere interna o esterna al documento XML.
 - Un'entità si dice *interna* quando il suo valore viene specificato all'interno del documento stesso.
 - Un'entità si dice *esterna* quando il suo valore viene specificato in un'unità di memorizzazione esterna al documento, ossia un file diverso dal documento XML.

Riferimenti alle entità

- Per includere il valore di un'entità all'interno di un documento XML è necessario inserire nel punto desiderato un riferimento all'entità.
- Un riferimento ad un'entità è costituito dal nome dell'entità preceduto da '&' e seguito da ';':

&NomeEntità.

Esempio di riferimento a entità

Un esempio: definita nel prologo l'entità colore:

```
<!ENTITY colore "rosso">
```

all'interno del documento ci si può riferire a tale entità in questo modo:

```
<bottone> il colore del bottone è &colore; </bottone>
```

Nel documento XML interpretato apparirà la stringa:

```
"il colore del bottone è rosso"
```

Document entity

Le componenti della “document entity” sono:

- elementi;
- attributi;
- entità.

Elementi

Un elemento è un'unità logica formata da:

- un tag di apertura;
- un contenuto;
- un tag di chiusura.

Apertura e chiusura di elementi

- Con *tag di apertura* si intende una parola, (il nome dell'elemento), racchiusa tra le parentesi angolari '<' e '>':

<nomeElemento>

- Con *tag di chiusura* si intende una parola, che rappresenta il nome dell'elemento, racchiusa tra i simboli '</' e '>':

</nomeElemento>

Contenuto di un elemento

- Il *contenuto* di un elemento è il testo contenuto tra i tag di definizione dell'elemento.
- Il contenuto può essere composto:
 - da altri elementi, detti figli (o child) dell'elemento iniziale (detto padre);
 - da una stringa di testo, che rappresenta l'informazione corrispondente all'elemento.

Element content

- Nel caso un elemento contenga solo altri elementi si dice che l'elemento ha un *element content*.

Esempio:

```
<nomeElemento>
```

```
  <nomeElementoFiglio1> ... </nomeElementoFiglio1>
```

```
  <nomeElementoFiglio2> ... </nomeElementoFiglio2>
```

```
</nomeElemento>
```

L'elemento `<nomeElemento>` ha element content in quanto contiene solo due elementi.

Elementi senza contenuto

- In XML possono esistere elementi senza contenuto.
- La sintassi prevede un unico tag, quello di apertura, contenente un numero a piacere di attributi e il simbolo '/' anteposto alla parentesi angolare di chiusura del tag:

```
<nomeElemento  
nomeAttributo="valoreAttributo" ... />
```

Elementi: significato dei tag

- XML non specifica il significato di tag ed attributi (a differenza di altri linguaggi di marcatura, come HTML) ma li utilizza solo per strutturare l'informazione e lascia l'interpretazione del documento all'applicazione che lo utilizza.
- Ad es. il tag `<p>`
 - in HTML indica un concetto deciso da chi scrive il documento paragrafo;
 - in XML può indicare un qualsiasi elemento XML.
- I commenti sono espressi come in HTML, ovvero con i simboli: `<!-- ... -->`

Attributi

- Un attributo è una coppia nome-valore che apporta un'informazione aggiuntiva all'elemento.
- Viene inserito all'interno del tag di apertura dell'elemento e deve essere scritto nella forma:

nomeAttributo="valoreAttributo"

- Ogni elemento può avere un numero di attributi maggiore o uguale a zero.

Esempio di attributi

- Un esempio dell'uso di XML potrebbe essere la descrizione dei clienti di un'ipotetica azienda:

```
<cliente id="23968">  
  <nome> azienda A. </nome>  
  <indirizzo nazione="italia">  
    <via> viale Mazzini n.34 </via>  
    <codice> 00196 </codice>  
    <citta> Roma </citta>  
  </indirizzo>  
</cliente>
```

Struttura, contenuto e forma

- In un documento XML si possono contraddistinguere struttura, contenuto e forma:
 - la *struttura* di un documento XML è data dal modo in cui gli elementi si annidano all'interno del documento stesso;
 - il *contenuto* di un documento XML è l'informazione racchiusa all'interno dei tag;
 - la *forma* di un documento XML è la presentazione del documento su di un certo dispositivo d'uscita.

XML è un meta-linguaggio

- Non definisce tag ma fornisce delle regole per poterli definire
- Offre una libertà estrema nella definizione dei tag di descrizione di un documento

“I documenti devono essere ben formati”

questo è un principio fondamentale che garantisce conformità e prevede alcune norme che i documenti devono rispettare per essere considerati corretti.

Correttezza e validità

Definizioni:

- Un documento conforme ad una *grammatica* viene ritenuto **valido**.
- Un documento non conforme, ma sintatticamente corretto, viene detto **well formed** (o **ben formato**), **non valido**.
 - Sintatticamente corretto.
 - Non rispondente alla grammatica definita.
- Per definizione, un documento valido è **well formed**.

Correttezza e validità

Un documento XML è **ben formato** se:

- ha un solo nodo radice o root, che contiene tutti gli altri elementi del documento.
 - Le parti che possono stare all'esterno di questo elemento sono i commenti e le direttive di elaborazione (per es., la dichiarazione della versione di XML);
- Ogni elemento deve avere un tag iniziale e uno finale-
 - Se vuoto, può prevedere la forma abbreviata (“/>>”).

Correttezza e validità

- Gli elementi devono essere appropriatamente annidati, cioè i tag di chiusura devono seguire l'ordine inverso dei rispettivi tag di apertura.
- I valori degli attributi devono essere racchiusi fra apici, singoli o doppi.
- XML è un linguaggio *case sensitive*.
 - I nomi dei tag e degli attributi devono coincidere nei tag di apertura e chiusura.

Utilizzo di documenti XML

- Lo standard XML non definisce solo il modo in cui scrivere i documenti XML ma anche il modo in cui debbano essere utilizzati.
- Per utilizzare un documento XML si deve avere a disposizione un XML processor e un'applicazione.
- XML processor (o parser) é un modulo software utilizzato per leggere un documento XML.
 - Fornisce accesso al contenuto e struttura del documento.

Validazione di documenti XML

- L'XML processor legge il documento XML e controlla sintassi e struttura logica in base alle regole fornite.
- Per verificare la validità del documento, l'XML processor deve conoscere il riferimento all'opportuna grammatica.
- La mancata corrispondenza tra documento XML e grammatica oppure la mancata indicazione della grammatica rende il documento non valido.
 - L'XML processor non genera un errore ma solo un messaggio d'avviso.
 - L'intero documento viene comunque messo a disposizione dell'applicazione.

Entità e simboli speciali

- XML definisce cinque elementi sintattici, detti entità, che devono essere utilizzati al posto dei corrispondenti caratteri:
 - & definisce il carattere &
 - < definisce il carattere <
 - > definisce il carattere >
 - " definisce il carattere “
 - ' definisce il carattere ‘

Esempio

- Per descrivere un tag HTML in un documento XML si deve scrivere:

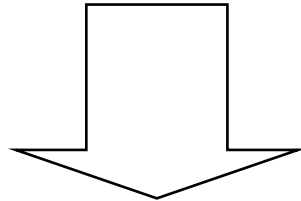
<descrizione>

Il tag <body> indica il corpo di un documento HTML...

</descrizione>

Come definire una grammatica

- XML consente di definire i tag liberamente, ma è necessario un meccanismo che vincoli il loro uso nei documenti stabilendo:
 - quali tag possono essere utilizzati
 - come devono essere utilizzati i tag per rispecchiare la struttura logica predefinita



È necessario definire una grammatica per il linguaggio basato su tag.

Document Type Definition (DTD)

- La grammatica deve indicare i vocaboli che si possono utilizzare (tag) e la struttura secondo cui si possono comporre le frasi (documenti).
- Un meccanismo per definire la grammatica siffatta è l'uso di Document Type Definition (DTD):
 - un documento che descrive la struttura logica di una classe di documenti XML specificandone entità, elementi ed attributi.

Documento validato da una DTD

- Un documento XML valido:
 - deve seguire le regole specificate nella DTD;
 - deve fornire al parser il modo per accedere a tali regole.
- In un documento XML il riferimento alla DTD, che può essere implicito o esplicito, viene specificato nella *dichiarazione del tipo di documento*.

Schemi XML

- L'approccio basato sulle DTD consente di specificare la struttura di un documento XML e di ciascun tag utilizzabile al suo interno.
- Quando c'è bisogno di un maggiore controllo sugli elementi che possono trovarsi all'interno di uno specifico tipo di documenti XML, le DTD non sono più sufficienti.
- Queste limitazioni possono essere superate utilizzando al posto dei DTD i cosiddetti **schemi XML** (o **XML Schema**).

Schemi XML

- Uno Schema XML associato a un documento XML svolge la stessa funzione di una DTD:
 - il documento XML può essere validato tramite il suo Schema.
- Lo schema XML offre strumenti più completi:
 - ad esempio per vincolare non solo l'annidamento o il tipo di ciascun elemento, ma anche il range di valori di ogni campo.
- Un altro vantaggio degli schemi XML è che sono scritti in XML.

Vantaggi degli schemi XML

Gli schemi XML sono:

- **più espressivi** delle DTD, causa alcune limitazioni, come la mancanza dei tipi di dati;
- **espressi in XML**, nella sintassi XML, aiutando gli sviluppatori nella creazione e nell'implementazione di applicazioni conformi agli stessi schemi XML;
- **auto-descrittivi**: uno schema XML è indipendente e la sua struttura corrisponde alla struttura del documento XML che viene definito.

Vantaggi degli schemi XML

- Utilizzabili da applicazioni che adottano XML:
 - gli schemi sono stati creati per essere il più possibile compatibili con le attuali applicazioni XML .
- Direttamente utilizzabili in Internet:
 - gli schemi sono nati con l'idea di essere utilizzati primariamente con applicazioni Web.
- Ottimizzati per l'interoperabilità:
 - devono funzionare con le attuali implementazioni XML e comprendere i meccanismi necessari per inserire e importare gli schemi l'uno nell'altro.

Gli schemi superano le DTD

- Le DTD sono definite utilizzando la grammatica EBNF (Extended Backus-Naur Form):
 - sintassi particolarmente complessa.
- Le DTD non consentono la definizione di tipi di dati.
 - Non si può avere un controllo accurato sul contenuto degli elementi di un documento XML né sul valore degli attributi.
- Gli schemi XML consentono di definire ambiti di validità (namespace) di una grammatica:
 - consentendo la presenza, in un documento XML, di tag definiti in schemi diversi;
 - Un documento XML può dipendere da un solo DTD che definisce tutta la grammatica applicabile.

Capitolo 19

Il sistema operativo

Gestione dei file

Newyork.bmp

Nome di file in DOS

New York Skyline.bmp

Nome di file in Windows

I nomi dei file servono alla loro esatta identificazione.

Nomi di file

I percorsi sono invece necessari per ritrovare i file nei vari supporti di memorizzazione.

C:\Lettere\Paolo.doc

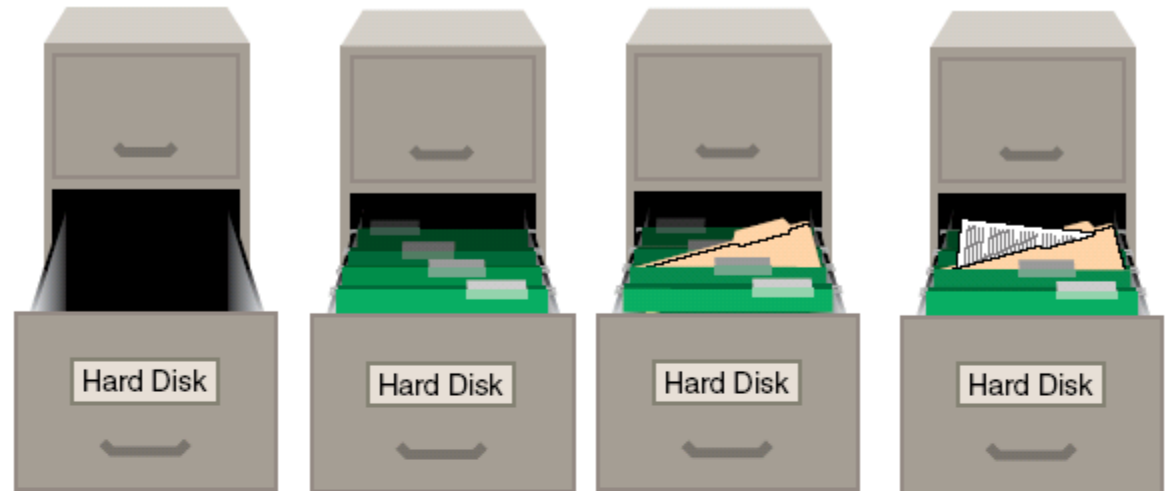
Il percorso del file in una cartella

C:\Lettere\Venditori\Paolo.doc

Il percorso del file in una sottocartella/cartella secondaria

Gestione delle cartelle

Le cartelle rappresentano le divisioni logiche dei supporti di memorizzazione: senza di esse, si avrebbe a che fare con un unico immenso spazio nel quale tutti i file sarebbero accatastati alla rinfusa.



Un disco fisso nuovo, come il cassetto vuoto di un archivio, non ha organizzazione.

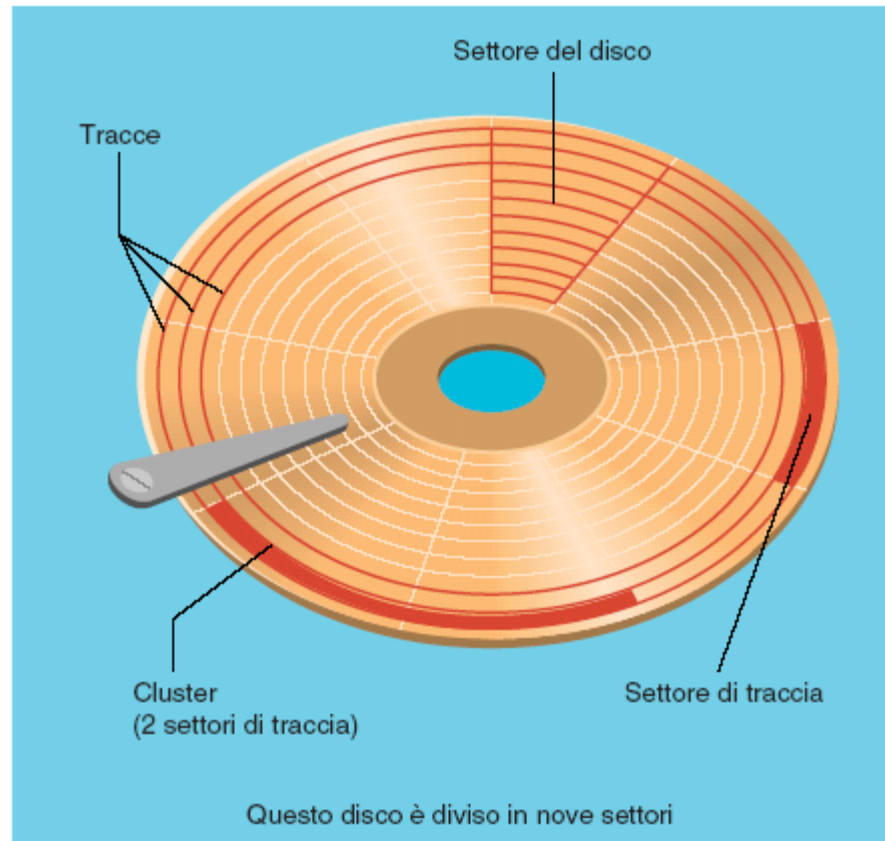
L'organizzazione del disco in directory o cartelle è paragonabile alla suddivisione del cassetto d'archivio tramite divisorii.

L'ulteriore suddivisione delle directory in sottodirectory, o delle cartelle in sottocartelle, è paragonabile all'aggiunta di ulteriori divisorii nel cassetto d'archivio.

I file possono essere memorizzati in qualsiasi directory o cartella.

Le cartelle

Formattazione dei dischi

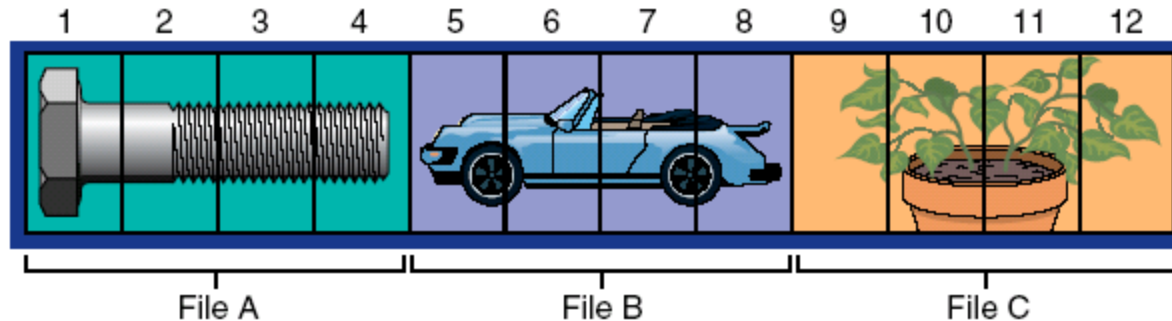


La formattazione del disco

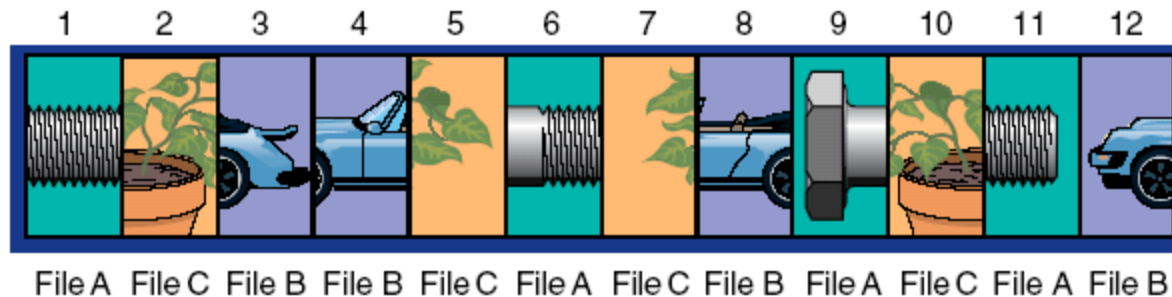
La formattazione prepara il supporto a ricevere i dati, suddividendolo in spazi (tracce, settori, cluster) idonei a ricevere i dati.

La deframmentazione

File non frammentati



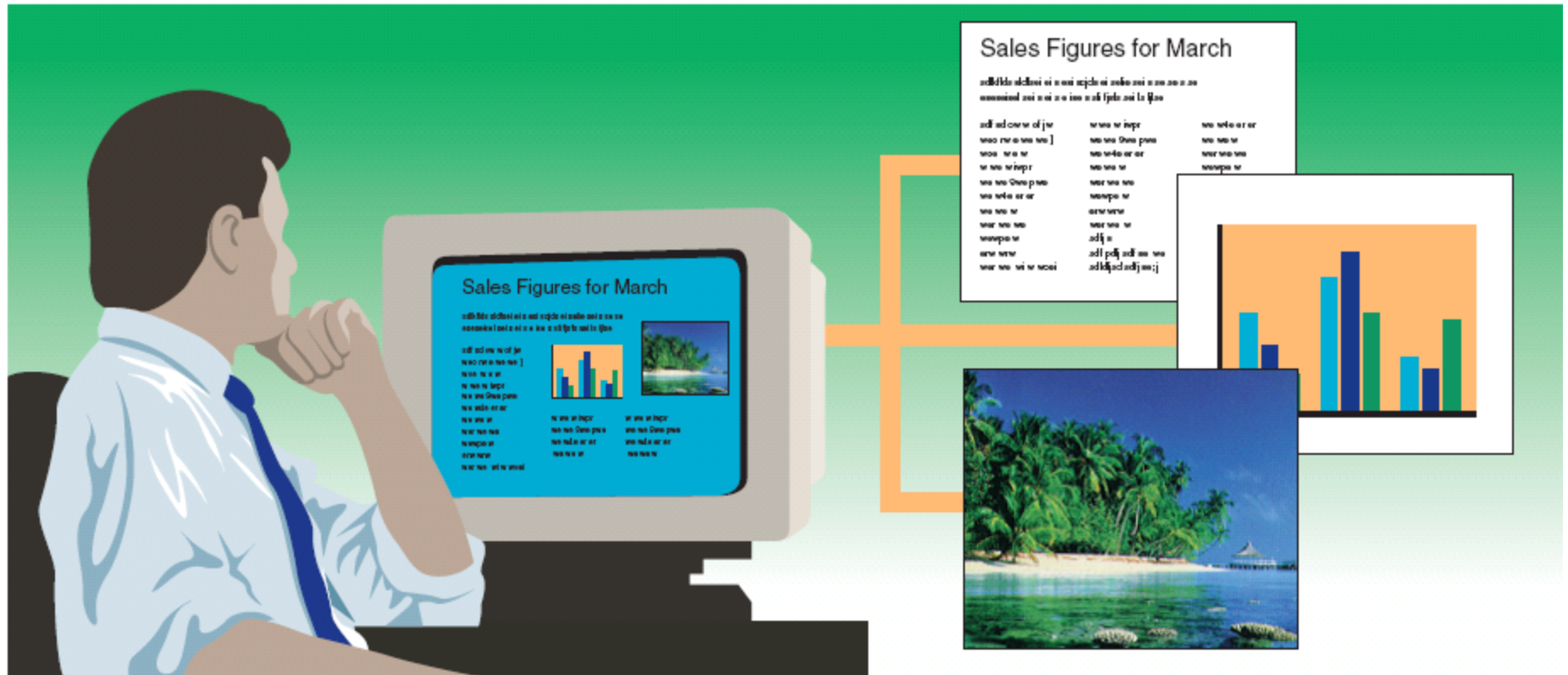
File frammentati



La frammentazione del disco

Non tutti i file sono memorizzati in spazi contigui, e il sistema operativo è costretto a un superlavoro per recuperarne tutte le parti sparpagliate sul supporto. La deframmentazione evita questo problema.

Incorporamento e collegamento



Il collegamento e l'incorporamento di oggetti in un documento

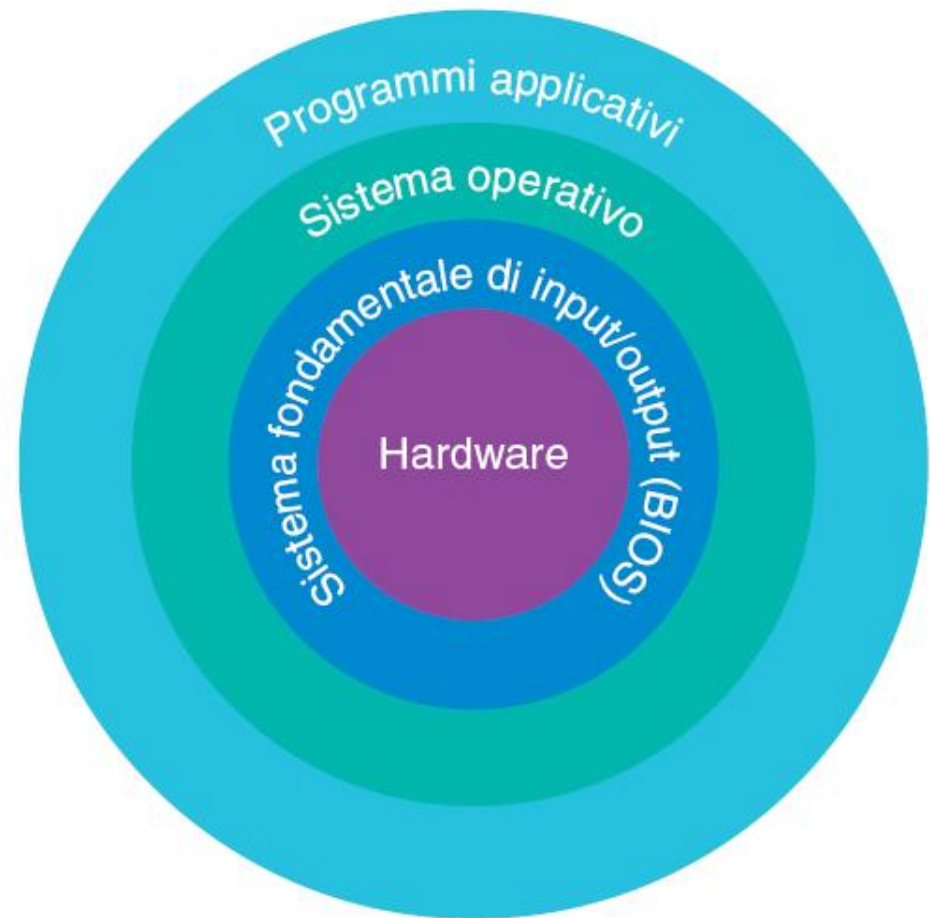
Si tratta di due tecniche con le quali si può far comparire in un unico documento la composizione di tutti o parti di altri documenti, non necessariamente dello stesso tipo.

Il sistema operativo: più in profondità



Livelli del software

La BIOS si rapporta direttamente con l'hardware del computer, fornendo informazioni al sistema operativo; i programmi applicativi gestiscono l'hardware solo attraverso il sistema operativo stesso.



Il software di base

- Il software di base comprende l'insieme dei programmi predisposti per un uso efficace ed efficiente del computer.
- Il sistema operativo:
 - fa parte del software di base;
 - è il gestore di tutte le risorse del computer.

Il sistema operativo (S.O.)

- Forma con l'hardware una struttura integrata del sistema di elaborazione dati.
- Può considerarsi come il coordinatore di tutte le attività rese disponibili dall'hardware.

Il sistema operativo

- Ha la funzione di astrarre, ad un livello di utilizzo più vicino alle abitudini dell'utente, la struttura hardware.
- Filtra le interazioni tra mondo esterno e computer.
- È il gestore delle risorse.

Il sistema operativo

La macchina virtuale è costituita da:

- hardware;
- firmware;
- assembler;
- sistema operativo;
- software applicativo.

Il S.O. gestore delle risorse

Sono risorse tutte le entità hardware e software che concorrono all'esecuzione dei programmi:

- memoria centrale;
- processore;
- memoria di massa;
- unità di I/O;
- dati e programmi.

Problematiche riguardanti le risorse

- Le risorse hanno caratteristiche estremamente differenti.
- Esempio: l'unità centrale è molto più veloce dell'unità memoria di massa (anche dell'ordine di 10.000 ad 1).
- Tali disparità possono comportare il degrado delle prestazioni del computer.

Modalità elaborative

- Multitasking
 - Windows, Linux, Mac, Unix, OS/2
- Monotasking
 - DOS

Modalità multitasking

- Un sistema di calcolo digitale può operare essenzialmente in:
 - modalità “batch processing” o “a lotti”;
 - modalità “interattiva”;
 - modalità “real time” o in “tempo reale”.

Sistemi operativi multitasking

- I sistemi operativi multitasking consentono di eseguire parallelamente più task (processi).
- Un task è un programma in esecuzione; rappresenta l'attività dinamica dell'esecuzione di un programma.

Il task

- Un task è definito da:
 - una componente statica:
 - il programma relativo;
 - una componente dinamica:
 - lo stato di avanzamento del task;
 - lo stato interno (i valori dei registri del processore);
 - la priorità;
 - altre informazioni.

Base del multitasking

L'idea che sta alla base del multitasking è la seguente: i task sono contemporaneamente nella RAM del Computer ed in modo concorrenziale ordinato condividono le risorse del sistema.

Code di processi

- In un determinato istante un processore esegue una sola operazione nelle modalità:
 - **mono e multitasking.**
- Si creano le “code di processi” (*process queues*).
- Ogni risorsa del computer gestisce una coda di processi.
- La coda della CPU viene chiamata coda dei pronti (*ready queue*).

Lo scheduler

- Lo scheduler ha il compito di selezionare da una coda un task.
- Long-term scheduler:
 - sceglie da un buffer dedicato i processi che devono passare nella coda dei pronti.
- Short-term scheduler:
 - sceglie dalla coda dei pronti i task che devono essere eseguiti dal processore.

Selezione per priorità

- Viene selezionato il task con priorità più alta.
- La priorità può essere statica o dinamica.
- I task più brevi hanno maggiore priorità (tempo di cpu).

Round Robin

- La *ready queue* (coda dei pronti) è vista come una coda circolare.
- Il processore (CPU) è assegnato, a rotazione, ai vari task in coda al massimo per il **time slice** (*quanto di tempo*).

Scelta del time slice

- L'efficienza del round robin dipende molto dall'ampiezza del time slice:
 - un time slice molto grande può rallentare;
 - un time slice molto piccolo fa intervenire spesso lo scheduler che appesantisce il sistema.

Modelli del S.O.: il modello onion-skin (1/2)

- Modello a strati
 - 1^ strato: **Hardware**
 - 2^ strato: **gestore CPU**
 - 3^ strato: **gestore Memoria Centrale**
 - 4^ strato: **periferiche virtuali**
 - (DRIVER logico): s/w che permette di accedere agli indirizzi delle periferiche
 - (DRIVER fisico): componente hardware
 - 5^ strato: **file system**

Modelli del S.O.:

il modello onion-skin (2/2)

- 6^a strato: **Programmi di libreria**
(editor, compilatori, interpreti, gestori di DB)
- 7^a strato: **Interfaccia grafica utente (GUI)**
 - Rendere visibile l'organizzazione dei file in memoria di massa
 - Facile lancio di programmi
 - Rendere friendly l'approccio uomo/macchina

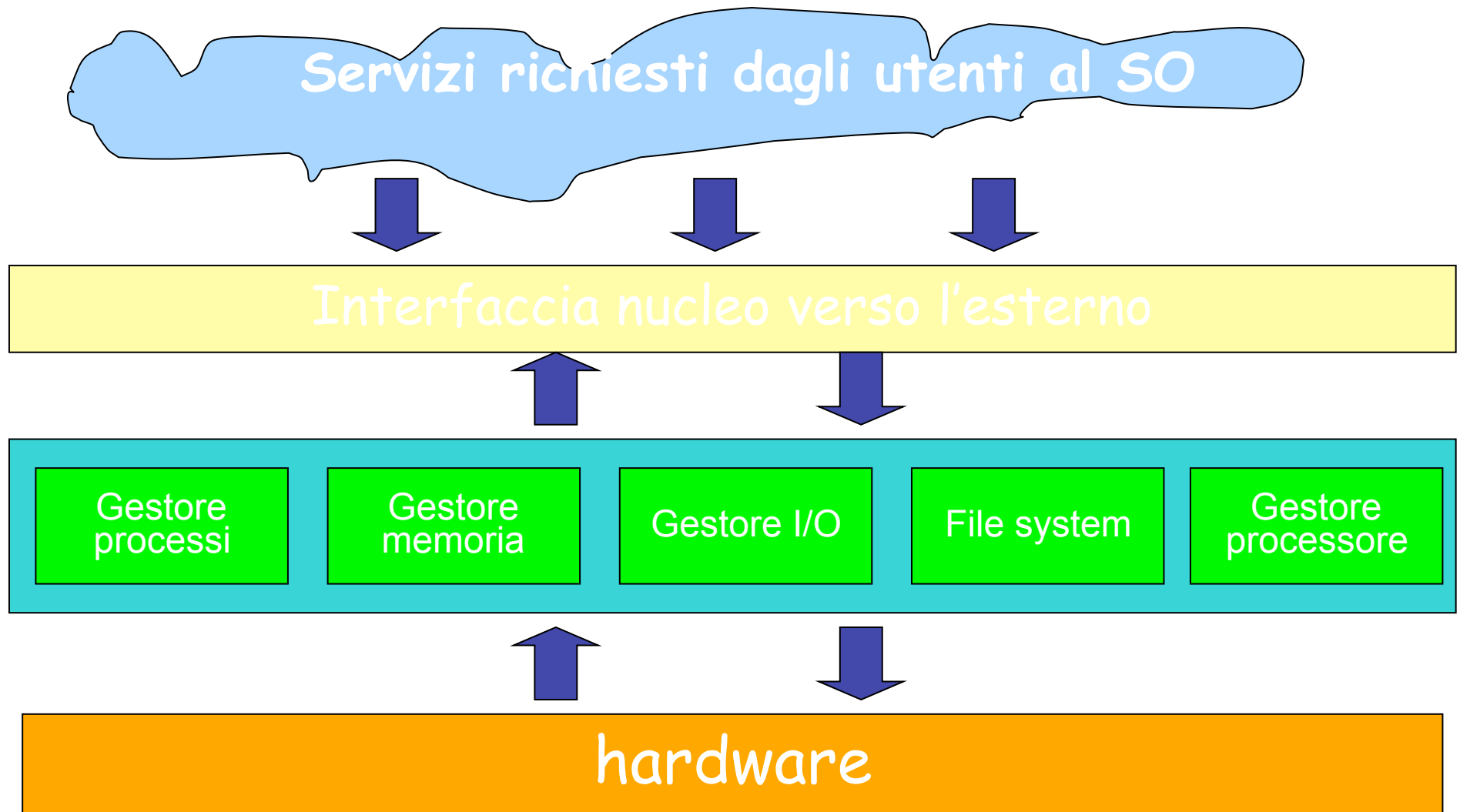
Funzionalità del sistema operativo



Sistema operativo: S/W di base

- Il sistema operativo è l'insieme di programmi che opera sul computer:
 - è organizzato a strati;
 - ogni strato rappresenta un livello di macchina virtuale.
- Nucleo

Struttura del S.O.: il nucleo



Funzionalità

- Possiamo identificare 5 strati tra macchina fisica e utente
 1. **Gestore dei processi**
 2. Gestore della memoria
 3. Gestore delle periferiche
 4. File system
 5. Interprete dei comandi

1. Gestore dei processi

- Gestisce l'esecuzione dei programmi.
- I programmi in esecuzione vengono chiamati processi (task) attivi nella CPU fino alla esecuzione completa.
- In ambiente multi utente il gestore (supervisore) decide a quale processo assegnare la CPU.
- Coordina gli eventi esterni (interruzioni provenienti dalle periferiche).

Un processo è...?

- Il ***programma*** è la parte **statica** rappresentata da una lista di istruzioni.
- Il ***processo*** è la parte **dinamica**:
 - programma tradotto e pronto per l'esecuzione.
- In un PC c'è un solo processo in esecuzione al tempo t^0 .

Funzionalità

- Possiamo identificare 5 strati tra macchina fisica e utente
 1. Gestore dei processi
 2. **Gestore della memoria**
 3. Gestore delle periferiche
 4. File system
 5. Interprete dei comandi

Il gestore della memoria centrale

- Il **gestore della memoria** (*memory manager*) è un componente del sistema operativo dedicato alla gestione della memoria RAM disponibile sul computer.
- Il gestore si preoccupa di **allocare**, **deallocare** e **gestire** la memoria che viene assegnata agli applicativi e allo stesso sistema operativo.

Il gestore della memoria centrale

- Il gestore della RAM si preoccupa di gestire ordinatamente i task che contemporaneamente risiedono in memoria centrale.
- I primi S.O. gestivano la memoria con lo swapping, oggi si utilizza la memoria virtuale.

Suddivisione della memoria

- La memoria centrale è suddivisa in (almeno) **due** partizioni:
 - partizione del **Sistema Operativo** Residente;
 - partizione dei **processi utente**.

2. Gestore della memoria

- Deve:
 - allocare la memoria (utilizzo da parte degli utenti);
 - partizionarla tra i processi che la richiedono .
- Contiene gli indirizzi degli strati di memoria dedicata ai processi.
- Si ha la rilocazione quando terminata la esecuzione di un task, il risultato viene depositato in uno spazio di memoria.

Rilocazione

- Statica:
 - il loader trasforma indirizzi **logici** in indirizzi **assoluti** durante il caricamento in memoria centrale.
- Dinamica:
 - il loader carica il programma con indirizzi logici (programma **compilato**) e mantiene gli indirizzi in memoria tramite un registro base.

Come opera il loader sulla RAM?

Attraverso la:

- segmentazione dello spazio di indirizzamento della RAM;
- paginazione;
- memoria virtuale.

Segmentazione

- La RAM viene suddivisa in segmenti di **lunghezza variabile (partizioni)** contenenti programmi e dati.
- È memorizzato il:
 - **registro base** (indirizzo della prima cella del programma in esecuzione);
 - **registro frontiero** (ultima cella del programma in esecuzione).
- Si può verificare la **frammentazione** della memoria:
 - interna;
 - esterna;
 - esempi di frammentazione;

La paginazione

- La memoria centrale e i programmi vengono suddivisi in pagine di **lunghezza fissa**:
 - i programmi vengono caricati in memoria **anche** in pagine **non contigue**;
 - la **tabella delle pagine** (**page table**) mantiene la corrispondenza tra pagine dei programmi in MM e pagine in RAM:
 - traduzione di un indirizzo logico nel corrispondente indirizzo fisico;
 - ogni pagina è costante ed è stabilita al momento della messa in opera del sistema operativo.

La paginazione

- Il task è suddiviso in tante partizioni (pagine di memoria) da piazzare fisicamente nella RAM.
- Il meccanismo della **page table** si occuperà della traduzione di un indirizzo logico nel corrispondente indirizzo fisico.
- La dimensione: ogni pagina è costante ed è stabilita al momento della messa in opera del sistema operativo.

La memoria virtuale

- È la tecnica adottata dai sistemi operativi multitasking attuali
- Il principio è il seguente:
 - ogni programma è diviso in blocchi di uguali dimensioni detti blocchi logici;
 - analogamente la RAM è divisa in pagine fisiche di dimensione uguale a quelle delle pagine logiche.

La memoria virtuale

- Con la suddivisione del programma in pagine **non** è necessario mantenere tutto il programma in memoria centrale...
- ...si utilizzano le tecniche di paginazione e segmentazione.
- Caratteristiche:
 - se la pagina del programma che contiene la successiva istruzione da eseguire non è in memoria si carica da disco (**page swapping**);
 - occorre trovare spazio in memoria centrale (si rilasciano le pagine già eseguite);
 - la tabella può contenere le pagine in ordine **sparso**.

La frammentazione della RAM: la memoria virtuale

- La tecnica della memoria virtuale elimina la frammentazione della RAM.
- Si utilizza la tecnica dello swapping per il caricamento dei programmi (task).

Swapping

- Lo swapping riguarda il salvataggio dello stato di un task nel momento in cui esso rilascia la C.P.U. (swap-out) per far posto ad un nuovo task (swap-in).

Più processi in memoria

- Avere più processi in memoria per applicare al meglio il ***multitasking***.
- Basterà creare più Running Region per ottenere più processi in memoria.

Frammentazione della RAM

- La frammentazione è un indicatore della quantità di memoria inutilizzabile a seguito di una particolare configurazione dei processi presenti nel sistema.
- Più alto è il tasso di frammentazione e peggiore è l'utilizzo del sistema.

Funzionalità

- Possiamo identificare 5 strati tra macchina fisica e utente
 1. Gestore dei processi
 2. Gestore della memoria
 3. **Gestore delle periferiche**
 4. File system
 5. Interprete dei comandi

3. Gestore delle periferiche

- Verifica le caratteristiche hardware delle periferiche.
- Gestisce le operazioni di input e output.
- Fornisce facility operative per l'utente.

Gestione delle periferiche

I dispositivi sono:

- driver fisici (hardware)
 - per trasferire e manipolare i dati;
- driver logici (software)
 - per funzioni di indirizzamento delle periferiche.

Driver fisico

- Controlla i meccanismi fisici dell' unità di lettura dei dispositivi esterni di I/O (device controller).
- Interagisce con la CPU attraverso i registri.
- Utilizza una memoria dedicata alle operazioni I/O (Memoria ad accesso diretto).

Driver logico

- Software che:
 - gestisce gli errori in lettura/scrittura;
 - gestisce i nomi/indirizzi del device driver;
 - gestisce le code di attesa.

Funzionalità

- Possiamo identificare 5 strati tra macchina fisica e utente
 1. Gestore dei processi
 2. Gestore della memoria
 3. Gestore delle periferiche
 4. **File system**
 5. Interprete dei comandi

4. File System

- Gestisce i dati in memoria di massa.
- I dati sono strutturati in modo gerarchico utilizzando *file* e *directory*.
- Fornisce utility per la gestione dei file (protegge i dati da accessi esterni).
- Garantisce la sicurezza dei dati.

Gestione dei file

- La memoria di massa memorizza grandi quantità di dati.
- Caratteristiche:
 - **persistenza**;
 - **sicurezza**.
- I dati vengono organizzati logicamente in file.

File

- Il file è un insieme omogeneo di dati.
- Nome:
 - identifica il file spesso con una estensione che indica il tipo di file;
 - es. Tesi.doc, somma.exe.
- Struttura:
 - sequenza di blocchi (record) di byte suddivisa in campi.
- Tipologia di accesso (sequenziale, diretto, ecc.).

Esempi di file in Windows

- **.exe** sono programmi eseguibili.
- **.txt** sono file di testo semplice.
- **.doc .xls .mdb .ppt** sono documenti rispettivamente di Word, Excel, Access, PowerPoint.
- **.htm e .html** sono pagine web.
- **.wav .mp3** sono file audio.
- **.jpg .gif .bmp .png** sono immagini.
- **.mov .avi** sono filmati.

Operazioni su file

- Il ***file system*** consente di:
 - creare, cancellare, spostare, recuperare, modificare documenti in memoria di massa (memoria persistente);
 - modificare gli elementi di un file;
 - rinominare i file.

Directory (cartella)

- File che contiene informazioni su altri file:
 - organizza i file (dati) in modo gerarchico;
 - contiene la lista dei nomi dei file collegati.
- Il *file system* ha una struttura ad **albero** costituito da:
 - Radice = si riferisce all'intero file system;
 - Nodi interni = directory;
 - Foglie = documenti/programmi.

Path-names (percorso)

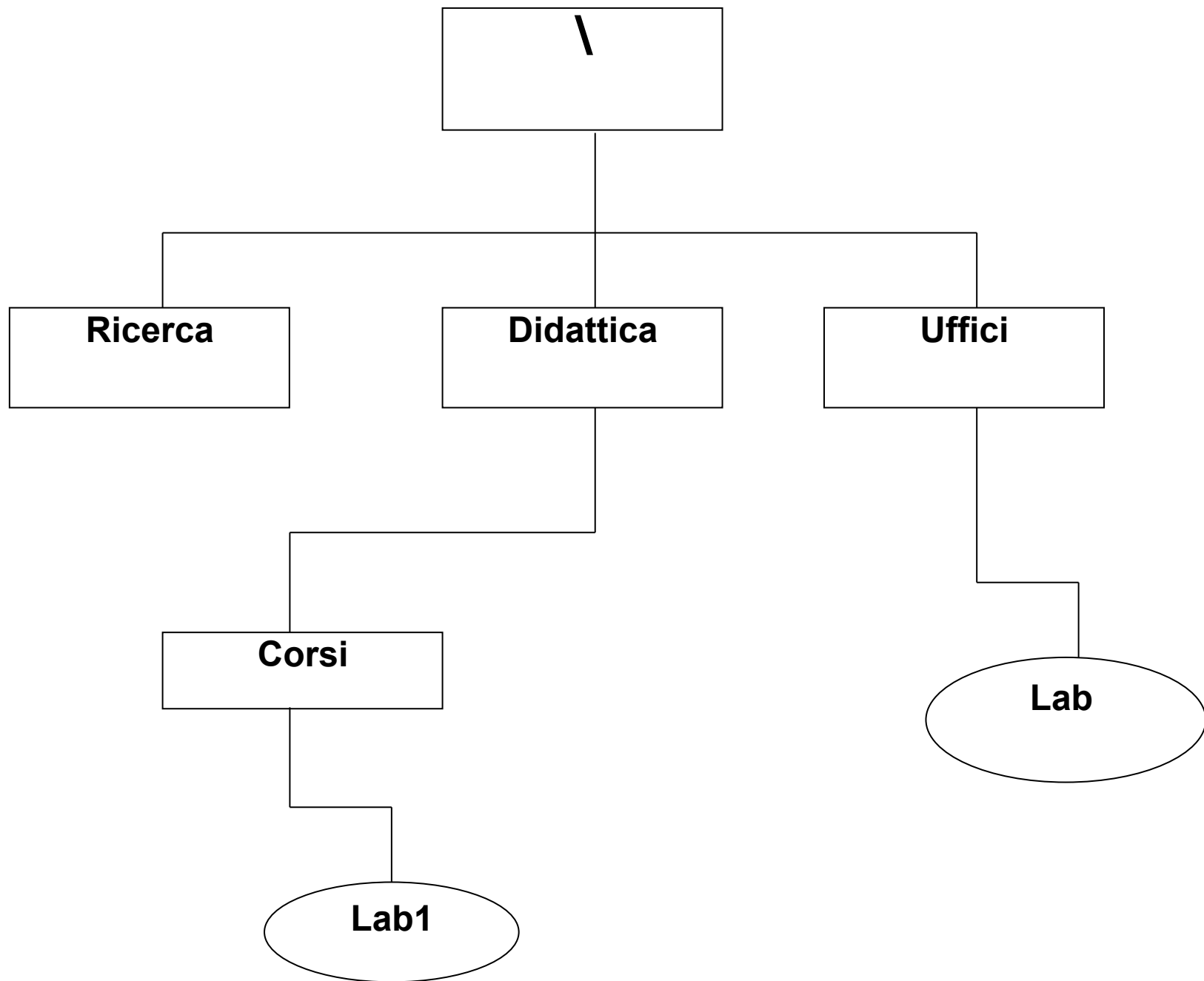
- Nella organizzazione ad albero specifica il percorso che si segue per ritrovare i corrispondenti file.
- Si identificano:
 - Nomi relativi:
 - relativi ad una particolare directory attiva
 - Nomi assoluti:
 - individua il cammino a partire dalla radice (root) del file system fino a raggiungere il file individuato

Il path-name

- Un file è individuato univocamente da un percorso (“pathname”) che indica l’intera sequenza di directory che lo contengono, a partire dal nodo “radice”.
- Il path-name di un file individua quel file specificandone la collocazione nella struttura gerarchica del file system, ovvero specifica il percorso che il sistema operativo deve percorrere per individuare il file.

Il path-name

- Il nome completo di un file è indicato con la notazione: ***percorso\nomefile***.
- Percorso → le directory da attraversare lungo l'albero per raggiungere la collocazione del file.
- \ → separatore tra directory.
- Nomefile → nome attribuito dall'utente per identificarne il contenuto o in automatico dal sistema operativo (il nome può contenere fino ad un massimo di 255 caratteri, spazi compresi).



Il path-name

Il percorso del file Lab1 sarà: **\Didattica
\Corsi\Lab1**

Il path-name

- Per fare riferimento ad un file, oltre al percorso, è necessario fare riferimento anche al supporto di memorizzazione (es. unità disco).
- Ogni unità disco è individuata da una lettera dell'alfabeto inglese seguito dal simbolo ":".
- In ambiente Windows le unità più frequenti sono le seguenti:
 - **A:** floppy disk;
 - **C:** disco rigido;
 - **D:** Cd-rom;
 - **E:**, **F:** ecc.: altre unità spesso rimovibili.

Il path-name

- Per cui, se il file Lab1 si trovasse sul disco fisso il suo path-name risulterebbe il seguente:

C:\Didattica\Corsi\Lab1

- Se invece si trovasse sul floppy disk, avremmo:

A:\Didattica\Corsi\Lab1

Il file system

- Per gestire il file system il Sistema Operativo Windows mette a disposizione due strumenti:
 - **risorse del computer;**
 - **esplora risorse.**

Gestione file system

- Tipologia di allocazione:
 - statica;
 - dinamica.
- I file possono essere memorizzati con:
 - struttura di blocchi di byte collegati (ogni blocco contiene l'indirizzo del successivo);
 - struttura con indice (si mantiene una tabella in memoria con gli indirizzi dei blocchi).

Funzionalità

- Possiamo identificare 5 strati tra macchina fisica e utente
 1. Gestore dei processi
 2. Gestore della memoria
 3. Gestore delle periferiche
 4. File system
 5. **Interprete dei comandi**

5. Interprete dei comandi

- Consente all'utente di lanciare i programmi.
- Contiene funzionalità per:
 - cercare in memoria il programma;
 - allocare la memoria richiesta dal programma;
 - lanciare una funzione per eseguire il programma.

Traduttore, linker e loader

- I traduttori di linguaggi procedurali sono
 - Compilatore, Interprete:
 - traducono programmi scritti in linguaggio ad alto livello in linguaggio macchina
 - Linker utilizzato per collegare i programmi divisi in moduli
- Loader
 - carica il programma compilato in memoria centrale

Utility di interfaccia utente

- Si tratta di programmi che consentono all'utente di interagire con il computer in modalità friendly.
- Esistono due tipologie di interfacce utente:
 - Interfaccia a linea di comando (DOS);
 - Interfaccia Grafica (GUI) windows.

Interfaccia a linea di comando

Rappresentazione testuale delle directory contenute nel disco residente.

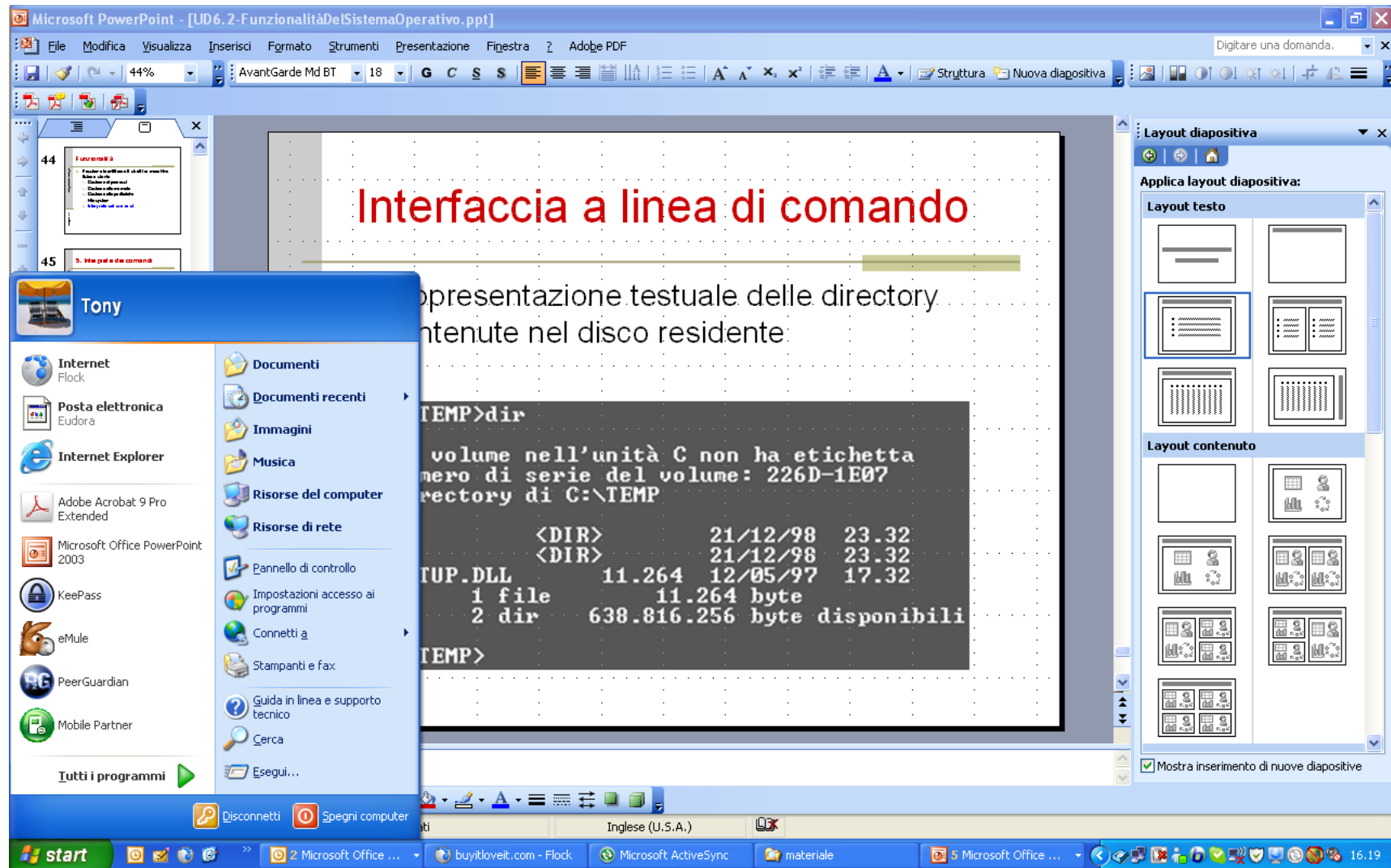
```
C:\TEMP>dir

Il volume nell'unità C non ha etichetta
Numero di serie del volume: 226D-1E07
Directory di C:\TEMP

.                <DIR>                21/12/98   23.32
..              <DIR>                21/12/98   23.32
_SETUP.DLL      11.264  12/05/97   17.32
      1 file           11.264 byte
      2 dir           638.816.256 byte disponibili

C:\TEMP>
```

Interfaccia grafica (GUI)

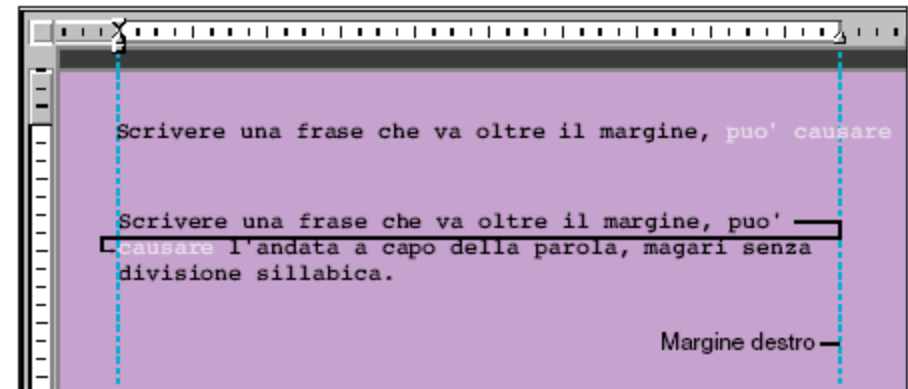


Capitolo 20

Al lavoro sui documenti

Funzioni di un wordprocessor (1/4)

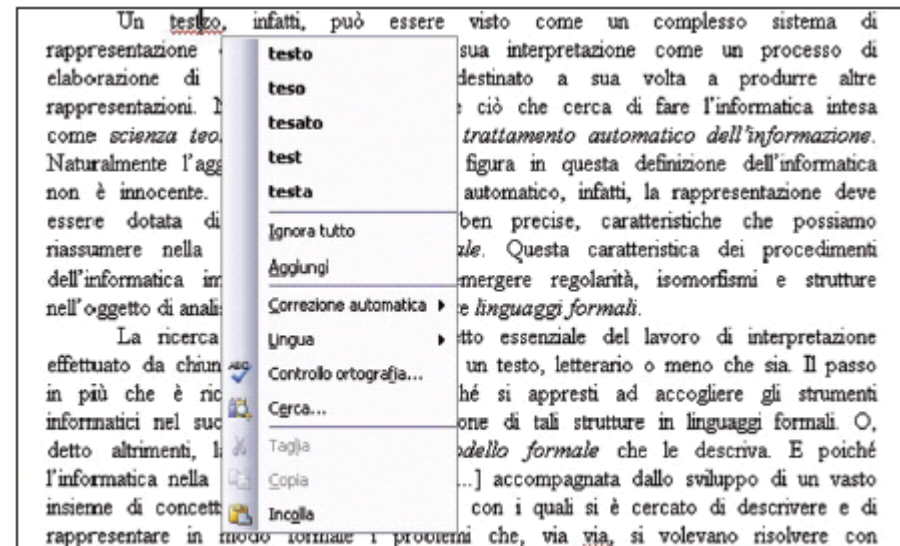
Le funzioni di un wordprocessor possono interessare la semplice digitazione del testo, in modo da renderla il più scorrevole possibile, che il contenuto di quanto scritto, evitando per esempio all'utente la fatica di controllare la propria ortografia.



La funzione "a capo automatico"

L'uso del tasto di ritorno (Backspace)	L'uso del tasto Cancella (Delete)
Backspace ←	Delete
Backspac e	← Delete
Backspa ce	De lete
Backsp ace	Del ete
Backs pace	Dele te
Back space	Delet e
Bac kspace	Delete
Ba ckspace	
B ackspace	
Backspace	

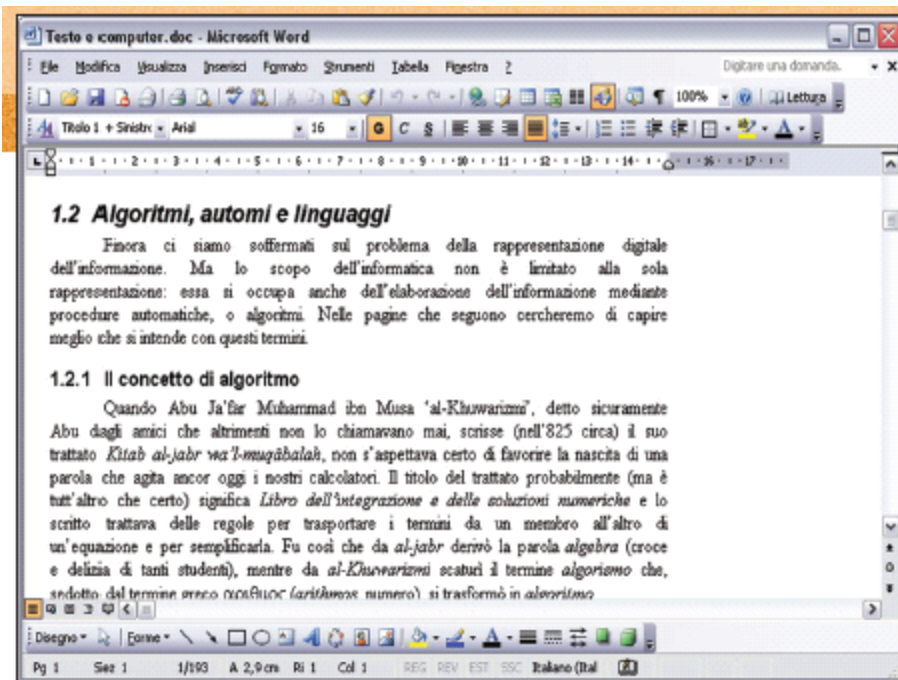
Le modifiche del testo



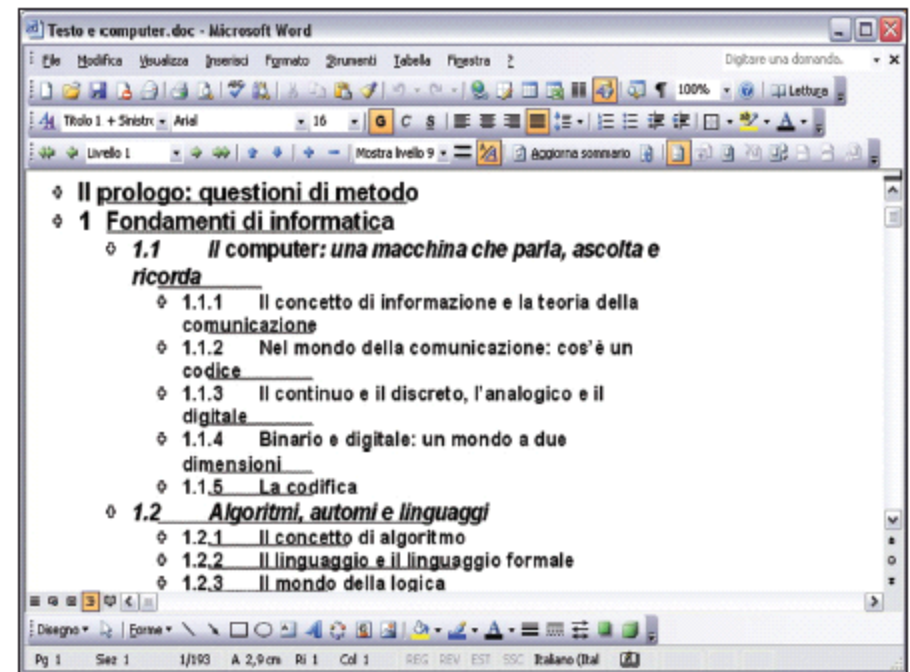
Il funzionamento del correttore ortografico

Funzioni di un wordprocessor (2/4)

Un wordprocessor permette di “vedere” il proprio documento in modi differenti. A sinistra il testo si presenta come verrà stampato, con in alto un righello millimetrato per rendersi conto delle effettive dimensioni, dei margini, etc. A destra, il testo riporta quelle strutture che una volta stampato sono invisibili: confini di paragrafi, elementi degli elenchi, delimitazioni per particolari scelte grafiche, e così via.



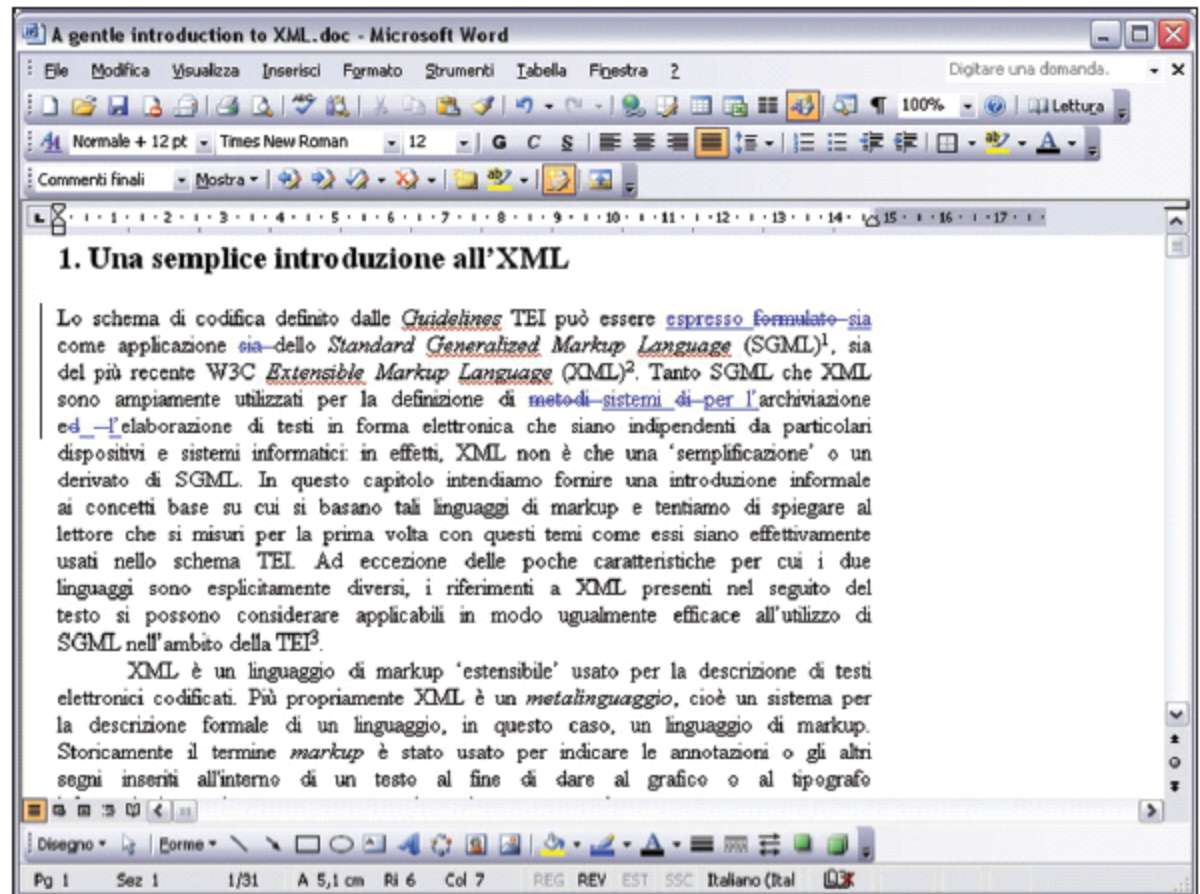
Visualizzazione normale



Visualizzazione struttura

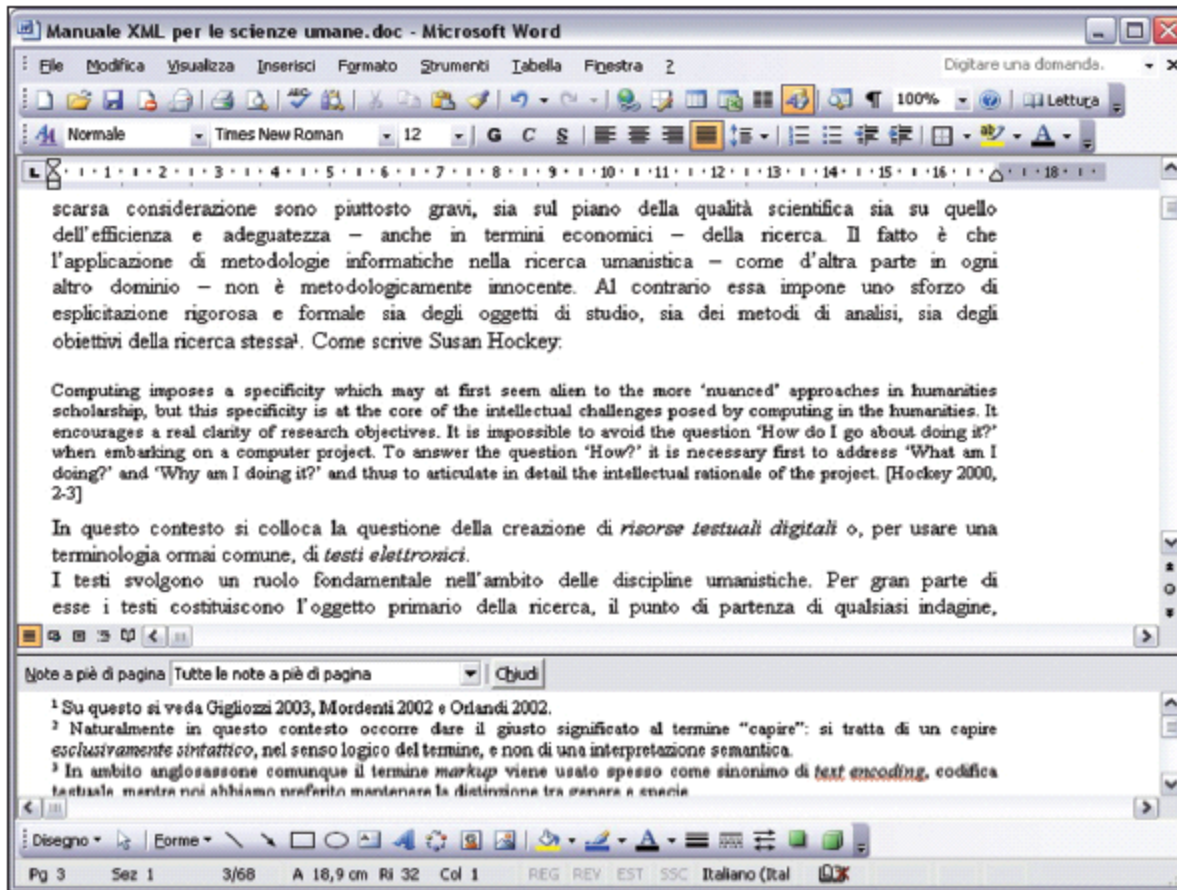
Funzioni di un wordprocessor (3/4)

La funzione che gestisce le revisioni di un documento è estremamente utile quando si lavora a più riprese sullo stesso testo, modificandolo (ad es., la tesi di laurea). Con questa funzione il wordprocessor può ricordarci quando abbiamo aggiunto o eliminato alcune parti del documento, in modo da poter ritornare sulle nostre decisioni o capire meglio le correzioni apportate da altri.



Gli indicatori di revisioni di un testo

Funzioni di un wordprocessor (4/4)



Un esempio di nota a piè di pagina

La gestione delle note è una funzione importantissima in un wordprocessor. Infatti le note rappresentano una parte “nobile” di un testo, che però spesso sono scritte in una zona diversa dal testo principale - a piè di pagina o a fine capitolo - e che richiedono una grande attenzione nella numerazione. Quindi il wordprocessor divide la sua finestra in sezioni diverse per mostrare sia la nota sia il testo cui si riferisce, e ne gestisce automaticamente la numerazione.

Fogli elettronici: immissione dati

*Le etichette
(corrispondenti alle celle
verdi) sono nomi descrittivi
che identificano un gruppo
di celle contenenti valori
numerici.*

	A	B
1	Rate mensili	
2	Calcolatore	
3	Costo dell'auto	€ 20.000
4	Acconto	€ 2.000
5		
6	Prestito richiesto	€ 18.000
7	Tasso d'interesse	12%
8	Durata (in mesi)	48
9		
10	Rata mensile	€ 474,01

*I valori numerici
(corrispondenti alle celle blu)
vengono utilizzati nelle formule
inserite in altre celle.*

*Le formule
(corrispondenti alle
celle gialle) svolgono
operazioni di addizione,
sottrazione,
moltiplicazione e
divisione.*

*Le funzioni
(corrispondenti alle
celle rosse) sono
formule complesse
predefinite.*

L'immissione dei dati nelle celle

Fogli elettronici: formule e formati (1/3)

	A	B
1	Vendite	€ 100
2	Costi	€ 90
3	Profitti	€ 10

Valori originali

	A	B
1	Vendite	€ 110
2	Costi	€ 90
3	Profitti	€ 20

Variazione delle vendite

	A	B
1	Vendite	€ 100
2	Costi	€ 95
3	Profitti	€ 5

Variazione dei costi

Le formule

Le formule permettono ai fogli elettronici di svolgere calcoli immediati, non appena l'utente inserisce i valori nelle celle di riferimento. In questo piccolo esempio, la cella B3 contiene la formula “=B1-B2”, che riporta istantaneamente il risultato della sottrazione.

	A	B
1	Numeri	Etichette
2	0,100099	Sinistra
3	0,10010	Centro
4	0,1001	Destra
5	0,100	Colore
6	0,10	Colore sfondo
7	10%	
8	€ 0,10	

I formati delle celle

Potendo contenere dati di qualunque tipo, i comandi di formato delle celle consentono di presentare i dati, numerici e testuali, in modi differenti per aspetto, allineamento, colore, valori di riferimento.

Fogli elettronici: formule e formati (2/3)

Più formule possono avere per oggetto lo stesso gruppo di dati, permettendo calcoli e rappresentazioni molto efficaci.

	A	B
1	Regione	Vendite
2	Est	10.000
3	Ovest	5.000
4	Centro	12.000
5	Sud	7.000
6	New England	5.000
7	Somma	39.000
8	Media	7.800
9	Minimo	5.000
10	Massimo	12.000
11	Deviazione standard	3.114,48

- =SOMMA (B2:B6)
- = MEDIA (B2:B6)
- = MIN (B2:B6)
- = MAX (B2:B6)
- = DEV.ST (B2:B6)

Le funzioni

Fogli elettronici: formule e formati (3/3)

Microsoft Excel - Fattura1

File Modifica Visualizza Inserisci Formato Strumenti Dati Finestra ?

Digitare una domanda.

Arial 10

L39

Q.tà	Descrizione	Prezzo unitario	TOTALE
1		€ 1.245,00	€ 1.245,00

Modalità di pagamento: Selezionare un'opzione...

Commenti: _____

Nome: _____

N. carta di credito: _____

Scadenza: _____

Imponibile: € 1.245,00

Spese di spedizione: _____

IVA: 10,00% € 124,50

TOTALE FATTURA

Solo per uso interno

Aliquota fiscale: Per aggiungere un'imposta o per modificarne la percentuale, immettere l'aliquota nella cella a sinistra. Il valore verrà calcolato automaticamente.

Combinando la potenza di calcolo con la duttilità grafica dei formati, è possibile ottenere da un foglio elettronico capacità molto complesse, come si vede da questo esempio di modello per fatture.

Un modello di fattura

Basi di dati (1/2)

Anche i database hanno delle funzioni grafiche e di formato che consentono a questi programmi di rendere più accattivante l'immissione dati e la loro visualizzazione.

The screenshot displays two Microsoft Access windows. The top window, titled 'Microsoft Access - [Dettagli ordini : Tabella]', shows a table with the following data:

ID ordine	Prodotto	Prezzo unitario	Quantità	Sconto
10248	Queso Cabrales	€ 26,00	12	0%
10248	Singaporean Hokkien Fried Mee	€ 55,00	10	0%

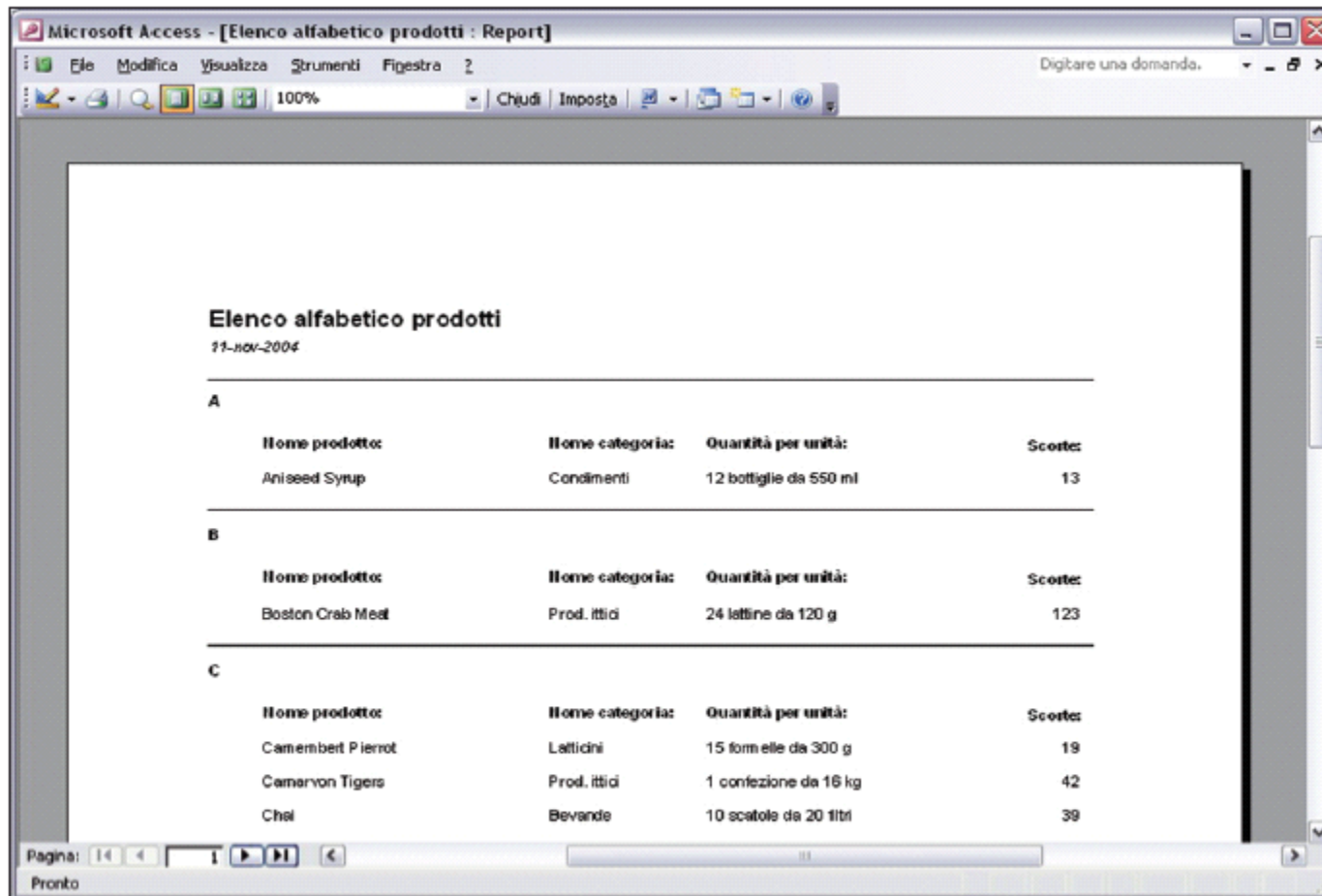
The bottom window, titled 'Microsoft Access - [Categorie]', shows a data entry form for food categories. The form includes the following fields and data:

- Nome categoria:** Condimenti
- Descrizione:** Salse dolci e piccanti, sottaceti, paste spalmabili e condimenti
- Immagine:** [Image of various condiments]
- Nome prodotto:** Aniseed Syrup
- Quantità per unità:** 12 bottiglie da 550 ml
- Prezzo unitario:** € 10,00
- Sospeso:** ☐
- Nome prodotto:** Chef Anton's Cajun Seasoning
- Quantità per unità:** 48 vasetti da 180 g
- Prezzo unitario:** € 22,00
- Sospeso:** ☐

A blue arrow points to the form, indicating its use for data entry.

Una maschera video per l'immissione dei dati

Basi di dati (2/2)



Microsoft Access - [Elenco alfabetico prodotti : Report]

Digitare una domanda.

100%

Chiedi Imposta

Elenco alfabetico prodotti
11-nov-2004

A

Nome prodotto:	Nome categoria:	Quantità per unità:	Scorte:
Aniseed Syrup	Condimenti	12 bottiglie da 550 ml	13

B

Nome prodotto:	Nome categoria:	Quantità per unità:	Scorte:
Boston Crab Meat	Prod. ittici	24 lattine da 120 g	123

C

Nome prodotto:	Nome categoria:	Quantità per unità:	Scorte:
Camembert Pierrot	Latticini	15 formelle da 300 g	19
Camervon Tigers	Prod. ittici	1 confezione da 16 kg	42
Chai	Bevande	10 scatole da 20 litri	39

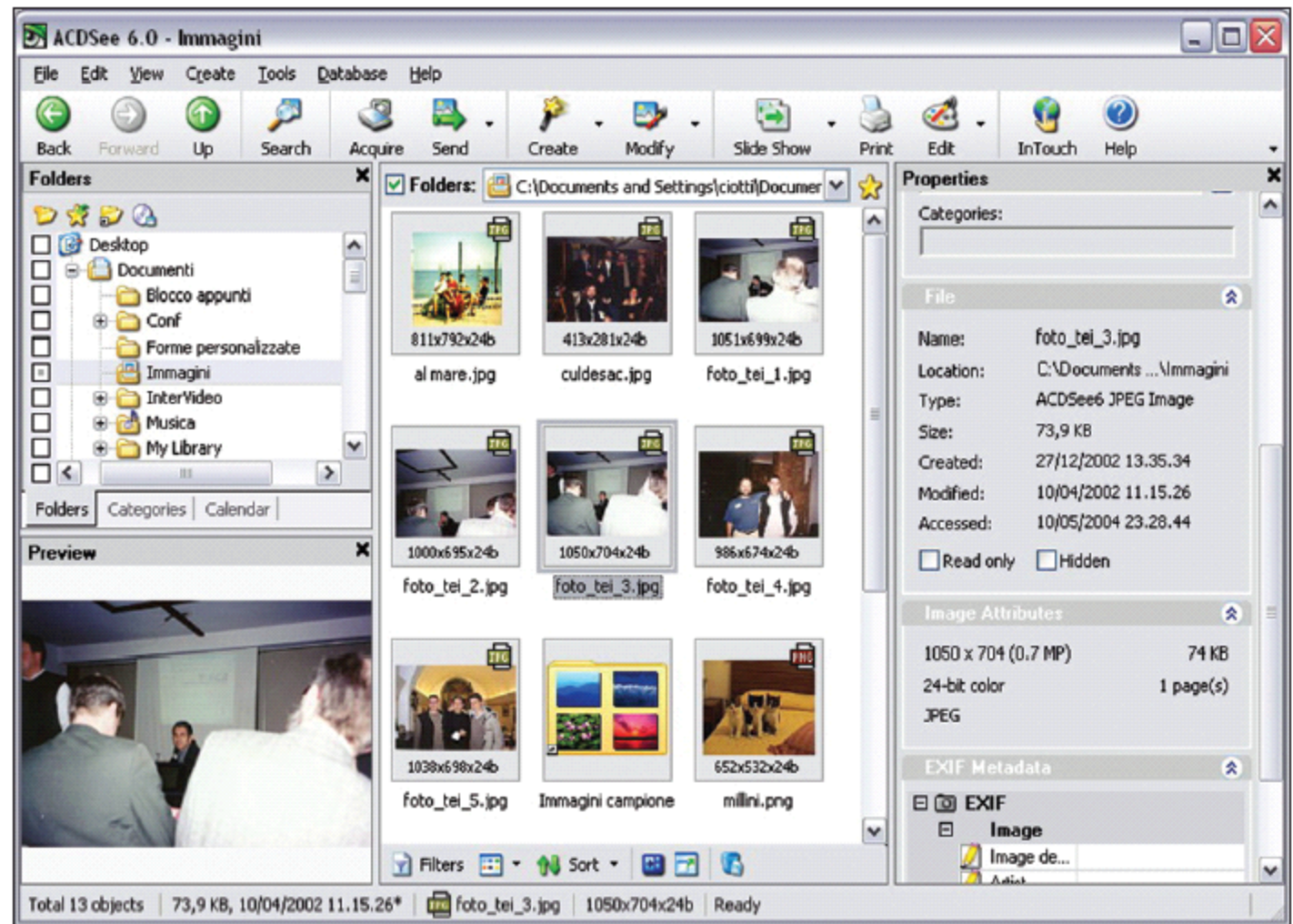
Pagina: 14
Pronto

Il risultato a stampa prodotto da un database può essere elaborato come un file prodotto di un wordprocessor, senza così perdere nulla in quanto a ordine, precisione e capacità grafiche.

Un report

Altri tipi di base di dati (1/2)

Anche programmi meno “seriosi” sono in realtà applicazioni che gestiscono basi di dati...



Un programma per la gestione dell'archivio fotografico personale

Altri tipi di base di dati (2/2)

...e tutte le pagine web nelle quali si scelgono articoli da comprare sono delle basi di dati.



Una base di dati on line per fare acquisti in rete – per gentile concessione di Eprice