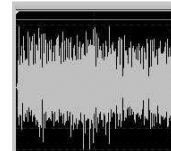


*Riassunto del corso e
cenni ad argomenti
"avanzati"*

In breve ...

```
<html>  
<body>  
.....  
</body>
```

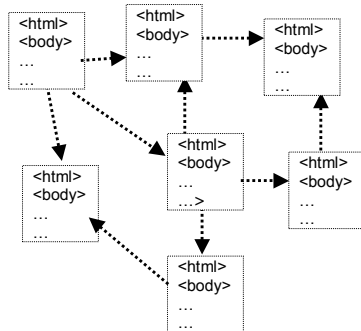


Sono tutte sequenze di bit!

```
□□□□□□□□  
1 1 0 1 0 1 0 0
```

```
□□  
0 1
```

In breve ...



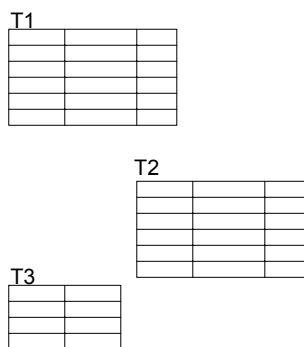
Iper testi in HTML

Informazioni di natura **non omogenea**, cui si accede mediante **legami associativi** realizzati tramite i link

Si naviga all'interno di grandi quantità di informazioni, secondo quanto viene in mente durante la fase di consultazione

3

In breve ...



Basi di dati

Informazioni di natura **omogenea e strutturata**, cui si accede mediante un linguaggio di interrogazione (**SQL**) basato sulla strutturazione logica dei dati

Schema concettuale
Schema logico

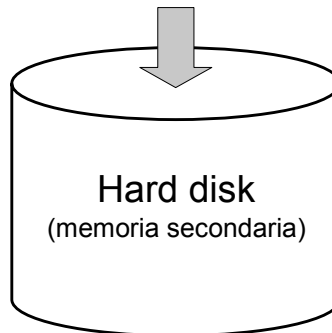
Schema fisico

4

In breve ...



Dato un documento di natura qualsiasi (testo, immagine, suono, base di dati), questo deve essere **memorizzato** in modo **permanente**

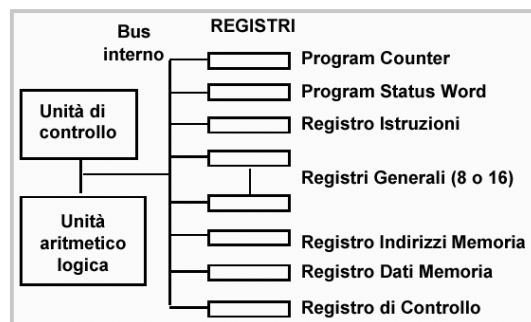


5

In breve ...

L'hard disk, oltre ai dati, contiene anche i **programmi** ...

Il **processore** è il componente del calcolatore in grado di **eseguire** i programmi

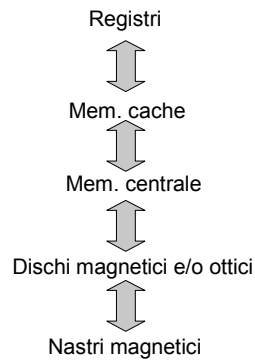


6

In breve ...

Durante l'esecuzione i programmi risiedono nella **memoria principale**

Possiamo definire una **gerarchia** di memorie

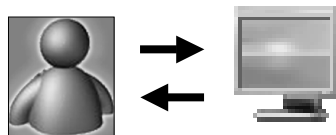


7

In breve ...

Processore e memorie da soli **non** bastano ...

Per interagire con il calcolatore ci vogliono anche i **dispositivi di input / output** che permettono di realizzare l'interazione uomo - macchina

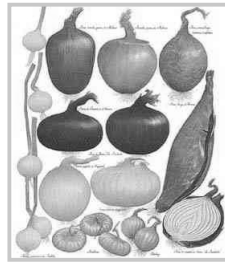


8

In breve ...

Processore, memorie e dispositivi di input / output formano l'architettura hardware dell'elaboratore ma da soli **non** bastano ...

È necessario introdurre il **software**, per creare quell'**ambiente virtuale** che l'utente usa senza dover conoscere i dettagli fisici dell'elaboratore



La struttura a cipolla!

9

In breve ...

Software di sistema

- ❖ avvio dell'elaboratore
- ❖ gestione dei processi
- ❖ gestione della memoria secondaria (file system)

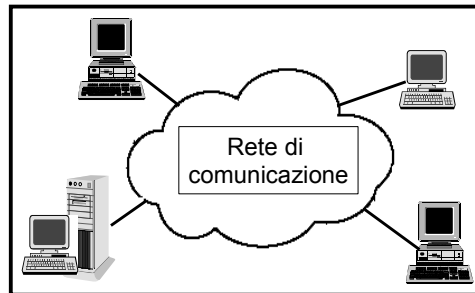
Software applicativo

- ❖ WordPad, Paint, Calcolatrice
- ❖ Netscape Navigator, Microsoft Internet Explorer
- ❖ PowerPoint, Word, Excel, Access

10

In breve ...

Dal computer isolato siamo passati alle **reti di computer** allo scopo di condividere risorse e dati

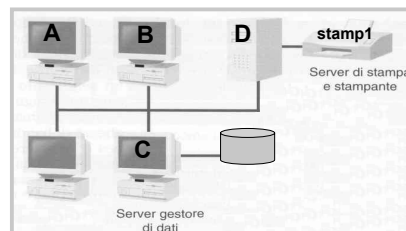


11

In breve ...

La gestione si complica e si deve introdurre il **sistema operativo di rete** per localizzare risorse e dati remoti

Meccanismo di interazione di tipo **client / server**



12

In breve ...

In questo modo l'utente può usare delle risorse che non sono direttamente collegate al suo elaboratore, in modo assolutamente **trasparente**



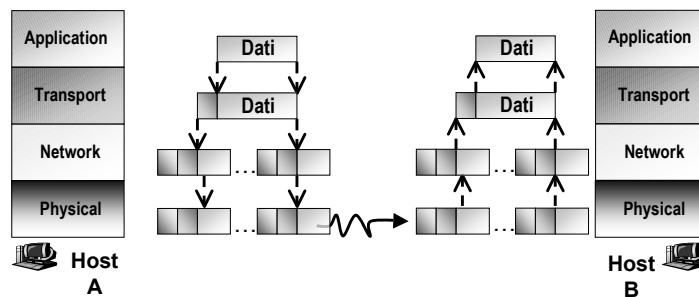
Interazione uomo – calcolatore remoto

13

In breve ...

Tante reti **eterogenee** collegate tra loro formano **Internet**

È necessario definire dei **protocolli di comunicazione** affinché gli elaboratori collegati in rete possano effettivamente comunicare tra loro

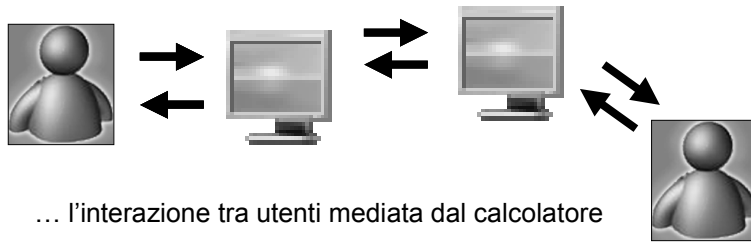


TCP/IP, protocolli che "tengono insieme" Internet

14

In breve ...

Internet sta cambiando il modo di comunicare, di studiare, di lavorare. Mediante i servizi che offre permette di realizzare ...



15

Cosa non abbiamo visto ...

Le ultime **evoluzioni dei linguaggi di markup**, ad esempio i fogli di stile (CSS) e il linguaggio XML

La **programmazione**, ovvero la possibilità di scrivere dei programmi per risolvere dei compiti specifici

Sistemi operativi o programmi applicativi "liberi", ovvero il software **open source**

Problematiche di **sicurezza, prestazioni, ...**

... e tantissimi altri aspetti dell'informatica

16

Evoluzione dei linguaggi di markup: **CSS**

I fogli di stile (Cascading Style Sheets - CSS) servono per facilitare la creazione di pagine HTML con un aspetto uniforme e personalizzato

Il Consorzio che si occupa della standardizzazione del linguaggio per il web (<http://www.w3c.org>) suggerisce di usare i fogli di stile perché facilitano la progettazione e la modifica delle pagine HTML

Permettono di **separare il contenuto** di un documento dagli aspetti legati alla sua **presentazione**

17

Evoluzione dei linguaggi di markup: **CSS**

dal sito www.html.it

.....
Ma cosa significa separare lo stile dai contenuti? Consideriamo un sito Web mediamente complesso, con un numero di pagine HTML pari a 100. Poniamo che il testo adottato per l'esposizione degli argomenti sia un "arial" corsivo. Con gli strumenti classici di HTML il codice per ottenere questo risultato e':

```
<font face="Arial"><i>Testo della pagina</i></font>
```

ripetuto in tutte le 100 pagine del sito, a chiusura e apertura del testo da formattare. Semplice, ovvio e per molti versi banale. Ma cosa succede nel momento in cui scegliamo di modificare il tipo di carattere a tutte le pagine? Non c'e' altra soluzione che aprire le 100 pagine e procedere ad altrettante modifiche, che sostituiscano il nome "Arial" con il nuovo font scelto. Un webmaster di medie capacita' puo' impiegare solo qualche decina di minuti, che diventano ore per siti di grandi dimensioni. Bene, questa perdita di tempo e' diretta conseguenza della promiscuita' tra stile e contenuto, laddove il primo (il tag FONT) non e' separato dal secondo (il testo della pagina). Viene da se' che la soluzione a questo problema e' nella separazione tra i due elementi sopracitati, che nella pratica si risolve adottando i Cascading style Sheets. D'ora in poi ci serviremo dell'acronimo CSS per richiamare questi "fogli di stile a cascata" che da tempo sono stati introdotti nel Web publishing, ma che solo negli ultimi mesi hanno conosciuto una grande diffusione.
.....

18

Evoluzione dei linguaggi di markup: XML

È un nuovo linguaggio di markup sviluppato dallo XML Working Group del W3C a partire dal 1996

Dal libro XML IN A NUTSHELL (O'REILLY)

"XML is one of the most important development in the history of computing. In the last few years it has been adopted in fields as diverse as law, aeronautics, finance, insurance, robotics, multimedia, hospitality, art, software design, physics, literature, ...

XML has become the syntax of choice for newly designed document formats across almost all computer applications."

19

Evoluzione dei linguaggi di markup: XML

Fino ad ora abbiamo visto che HTML fornisce un insieme **predefinito** di tag che specificano **come** dovrà essere visualizzata una pagina una volta letta da un browser

XML, invece, permette all'utente di definire il **proprio insieme di tag** (elementi)

I tag definiti dall'utente possono avere dei **nomi** che **rispecchiano il contenuto** del documento

20

Evoluzione dei linguaggi di markup: XML

```
<?xml version="1.0" standalone="no" ?>
<mailbox>
<memo>
  <from>
    <name>Rossi</name>
    <email>rossi@tin.it</email>
  </from>
  <to>
    <name>Verdi</name>
    <email>verdi@tiscalinet.it</email>
  </to>
  <subject>Esempio in XML</subject>
  <body>
    <par>Ti scrivo per avere delle spiegazioni su XML</par>
    <par>Ho letto da qualche parte .....</par>
  </body>
</memo>
.....
.....
</mailbox>
```

21

Evoluzione dei linguaggi di markup: XML

Le regole per la **visualizzazione** di un documento XML non sono scritte all'interno del documento stesso ma stanno su un **file a parte** che specifica, per ogni tag, come esso dovrà essere visualizzato

La separazione tra contenuto e regole di presentazione ci porta verso la **comunicazione multimodale**

22

Evoluzione dei linguaggi di markup: XML

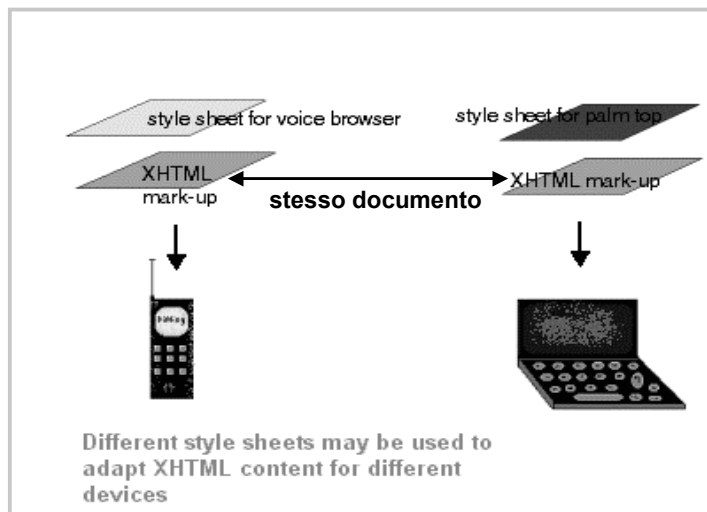


immagine presa dal documento <http://www.w3.org/style/Activity>

23

Programmazione

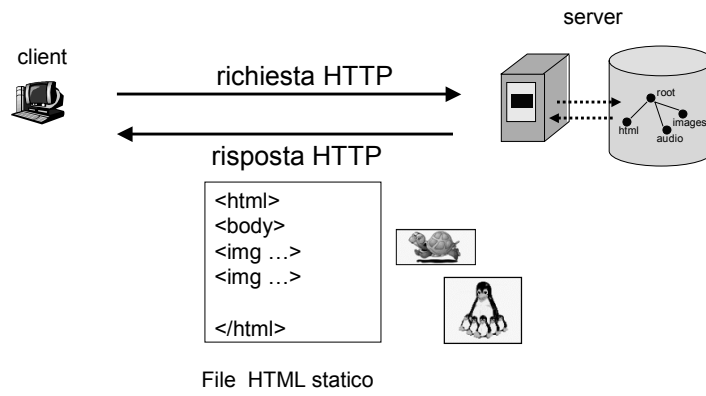
Abbiamo usato il laboratorio dei programmi del sistema operativo e dei programmi applicativi ma non abbiamo mai visto come si **scrivono** i programmi

I programmi sono scritti usando i **linguaggi di programmazione** formati da parole chiave (vocabolario) e da regole grammaticali che specificano come combinare insieme le parole del linguaggio

Non vedremo nulla di programmazione se non un semplice esempio per spiegare come un programma possa mettere in comunicazione i due "mondi" che abbiamo visto a lezione: le pagine HTML fruibili tramite un browser e le basi di dati

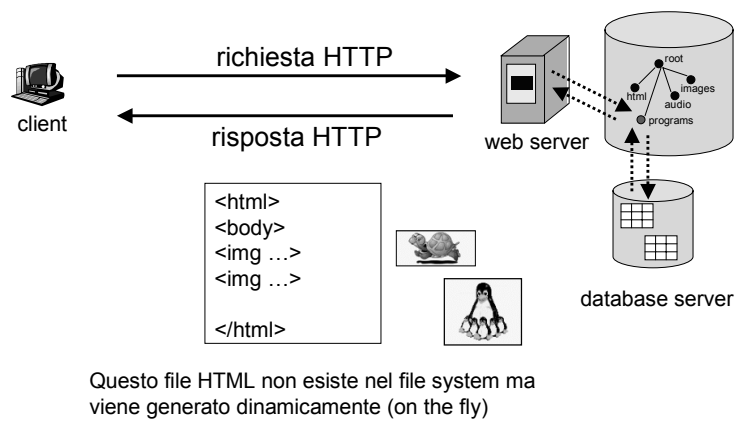
24

Programmazione per il web



25

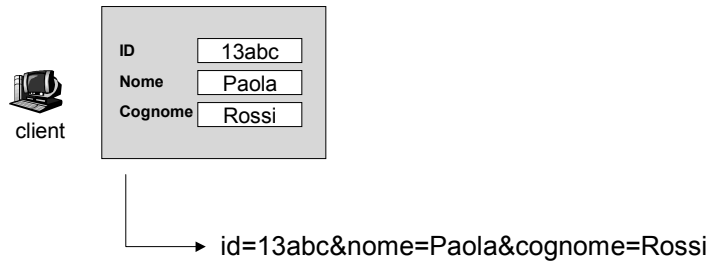
Programmazione per il web



26

Programmazione per il web

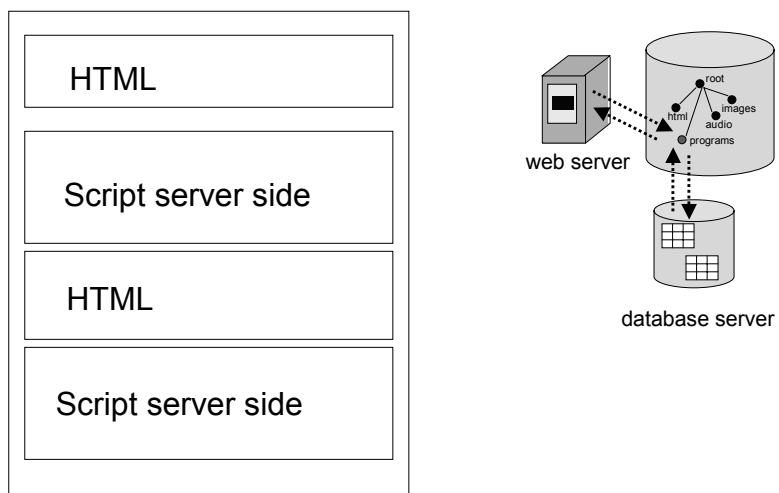
<FORM>



27

Programmazione per il web

Script e query SQL



28

Programmazione per il web

Script e query SQL

Read input from client

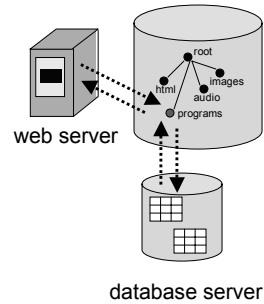
Create query

```
SELECT * FROM clienti WHERE ID=13abc
```

Connect to database and execute query

13abc
-------	-------	-------	-------	-------

Return data into an HTML file to the client



29

Programmazione per il web

```
<HTML>
<BODY bgcolor="white" text="red">

<%
'Dichiarazione delle variabili e reperimento delle informazioni in arrivo dal form
user_form = request.form("utente")

'Query di estrazione dei record relativo all'utente dato il numero di matricola
SQL ="select * from utenti where (matricola = " & user_form & " );"

'Connessione al DB
CurDir = Server.MapPath("../corso.mdb")
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open "DBQ=" & CurDir & ";Driver={Microsoft Access Driver (*.mdb)};"
Set cmdDC = Server.CreateObject("ADODB.Command")
Set rs = Server.CreateObject("ADODB.Recordset")
Set Rs=conn.Execute(SQL)

'Se il numero di matricola non appare nel DB
if (Rs.EOF) then
  Response.Write "<B>Accesso negato</B>"
else
'Se il numero di matricola appare nel DB
  Response.Write "<B>Accesso OK</B>"
end if
%>

</BODY>
</HTML>
```

Esempio di
script server side
in ASP

30

Oltre Microsoft?



31

Open source ☺



32

