

INFORMATICA DI BASE

- 6 crediti -

Docente: Michele Piana

Email: michele.piana@univr.it

URL: <http://www.di.univr.it/~piana>

Ricevimento: Lunedì ore 14:00 – 15:00

Studio: 2.05 (Ufficio del Preside)

RIASSUNTO DELLA LEZIONE PRECEDENTE

CONTEXT SWAPPING - I

Domanda cruciale: come avviene il passaggio di stato tra un processo e l'altro?

Definizione: si dice contesto di un processo, l'insieme dei dati che rappresentano lo stato di un processo: situazione della memoria, contenuto dei registri, livello di priorit ...

CONTEXT SWAPPING - II

Transizione di stato per un processo:

- **Quando un processo viene interrotto (ovvero esce dal suo stato di esecuzione), il sistema operativo salva il suo contesto in una struttura della memoria centrale detta descrittore del processo**
- **Quando il processo viene ripreso (torna in esecuzione) il sistema operativo provvede a ripristinare il suo contesto recuperando i dati dal descrittore**

CONTEXT SWAPPING - III

Il 'context swapping' e' il fenomeno per cui un processo P1 in esecuzione viene sostituito da un processo P2 pronto

Il SO provvede a:

- salvare il contesto di P1 e gestirne l'evoluzione (da attesa a pronto)**
- ripristinare il contesto di P2**

PROBLEMI: STARVATION

Diversi processi potrebbero accedere alla stessa risorsa ma solo alcuni di essi vi riescono, a causa della politica di gestione praticata dal sistema operativo

Esempio 1: il sistema operativo decide di assegnare la CPU sempre al processo con priorit  piu' alta. Ne segue che un processo con priorit  bassa non puo' mai accedere alla CPU

Esempio 2: assegnare ai tesisti la precedenza nell'uso delle stampanti: nei periodi di tesi i non-tesisti non stamperebbero mai

PROBLEMI: BLOCCO CRITICO

Un intero gruppo di processi rimane bloccato senza che la risorsa da loro contesa venga utilizzata da alcuno di essi

Esempio: il processo A detiene la stampante. Il processo A richiede i risultati del processo B per poter terminare le elaborazioni e, analogamente il processo B attende i risultati di A. Ne consegue che ne' A ne' B utilizzeranno la stampante (vincolo circolare)

SWAPPING

Nei sistemi a multiprogrammazione la memoria centrale non ha mai dimensioni tali da poter contenere tutti i programmi da eseguire in modo concorrente

Nella memoria di massa c'è un'area di swap in cui il SO trasferisce parte del contenuto della memoria centrale

Nello swapping vengono trasferiti su disco i dati relativi a processi in stato di attesa o di pronto

Gestione periferiche I/O

- Comandi **ad alto livello** per accedere alle periferiche che usano meccanismi quali:
 - **i controller**, dispositivi hardware per effettuare le operazioni di trasferimento dati;
 - dipendono dalle caratteristiche fisiche delle periferiche che gestiscono
 - l'interfaccia per la gestione di un mouse è sicuramente diversa da quella utilizzata per controllare il funzionamento di un lettore di CD-ROM.
 - **i driver**, programmi software per la gestione delle periferiche;
 - mascherano le caratteristiche specifiche dei controller,
 - forniscono un insieme di primitive ad alto livello per la gestione delle operazioni di ingresso/uscita utilizzabili dai programmi applicativi e dagli utenti.
- I sistemi operativi comprendono i driver per la gestione delle periferiche più comuni:
 - tastiera, video, mouse, ...
 - stampanti, scanner, ...
- Ogni aggiunta o modifica alla configurazione standard comporta l'installazione di software aggiuntivo (driver aggiuntivi).

Gestione memoria di massa

- **Obiettivo:**
presentare all'utente l'organizzazione logica dei dati e le operazioni che è possibile compiere su di essi.
- Operazioni di base di un file system:
 - **recupero** di dati precedentemente memorizzati;
 - **eliminazione (cancellazione)** di dati obsoleti;
 - **modifica/aggiornamento** di dati preesistenti;
 - **copia** di dati (e.g. da HD a FD) per backup o per il trasferimento;
 - ...
- I servizi vengono forniti sia ai **programmi applicativi** che direttamente agli **utenti**.

File system

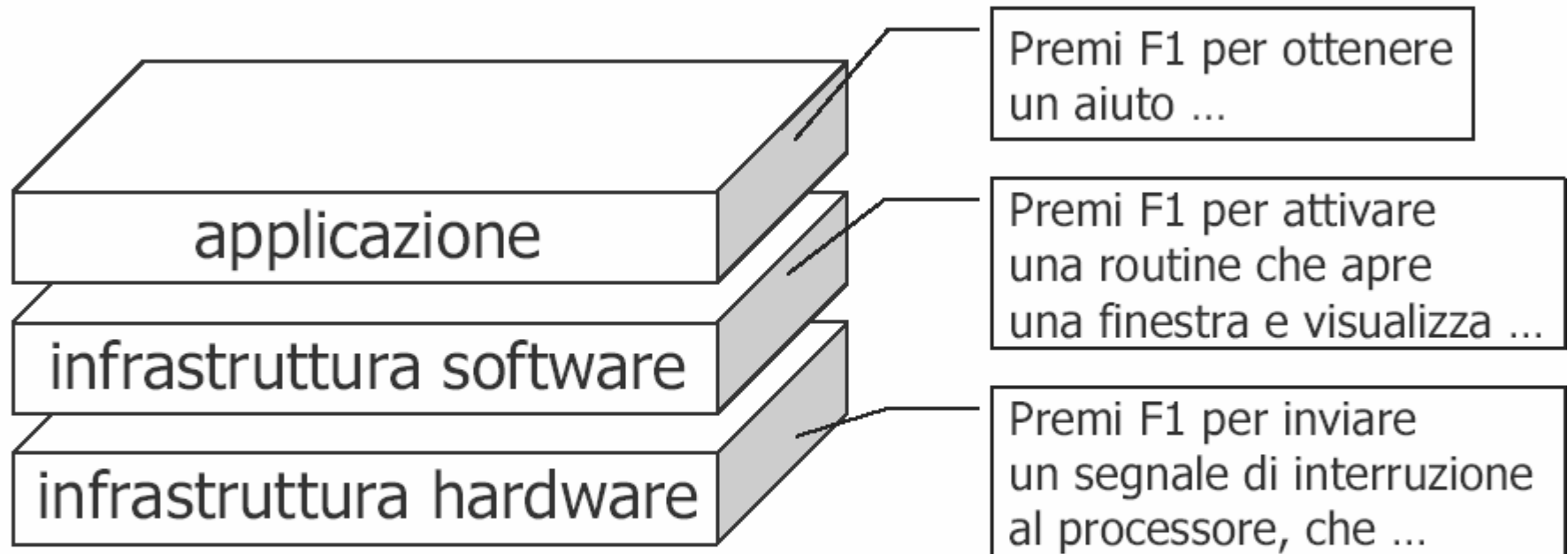
➤ **FILE:**

- contenitore logico di informazioni (dati o istruzioni);
- oggetto a “lunga vita”, per conservare le informazioni anche dopo la terminazione del processo che lo ha generato.

➤ Per ogni file:

- Identificatore (**nome.estensione**)
- Periferica (**drive**) e percorso sulla periferica
- Data creazione
- Dimensione
- Posizione effettiva dei dati nella memoria di massa
- Altre informazioni
 - applicazione che consente all'utente di “usare” il file
 - data di ultima modifica
 - diritti di accesso al contenuto del file
 - ...

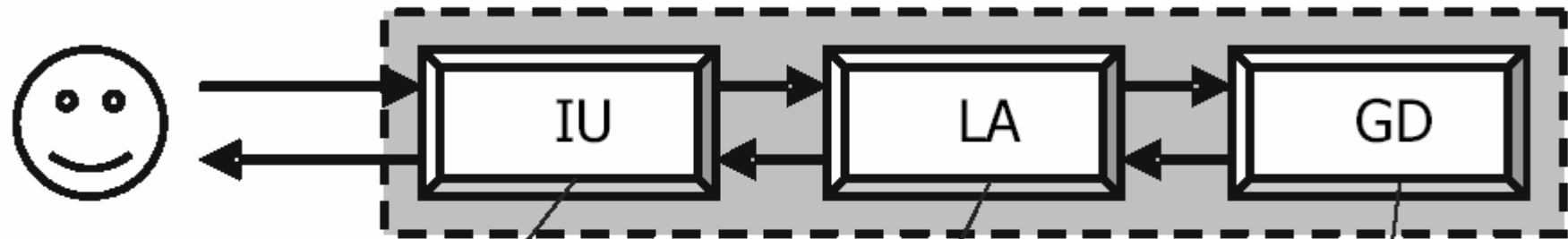
L'organizzazione "per livelli" dei sistemi informatici



... sono "punti di vista" diversi, e complementari, sullo stesso sistema...

Un'applicazione ben costruita (e funzionante...!) può essere usata prescindendo in larga misura dalle problematiche infrastrutturali dei livelli sottostanti

L'organizzazione funzionale di un'applicazione informatica

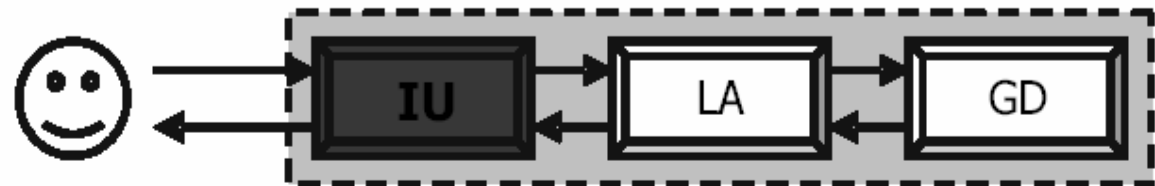


sottosistema di interfaccia con l'utente
(*user interface*, o anche *presentation layer*), incaricato di controllare i comandi e i dati che l'utente fornisce come input al sistema e di presentare come output i risultati dell'esecuzione del programma

sottosistema di logica applicativa
(*business logic layer*), che implementa gli specifici algoritmi di manipolazione dei dati che caratterizzano l'applicazione

sottosistema di gestione dei dati
(*data layer*), che si occupa dell'organizzazione dei dati, e in particolare della loro memorizzazione e del loro reperimento efficiente

Il sottosistema IU – interfaccia utente

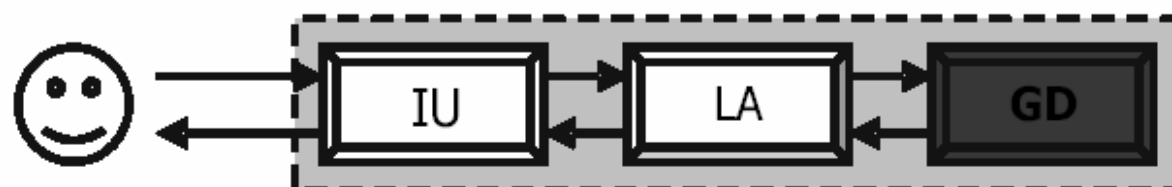


Ha il compito di gestire l'interazione del programma con l'utente:

- consente all'utente di specificare gli input al programma
 - restituisce all'utente gli output del programma
- e inoltre:
- fornisce informazioni sullo stato di esecuzione del programma

A causa dell'indipendenza funzionale tra sottosistemi,
le stesse caratteristiche per LA e GD possono essere ottenute
con IU diverse...

Il sottosistema GD – gestione dei dati



Ha il compito di organizzare i dati gestiti dall'applicazione, in particolare relativamente alla loro memorizzazione e al loro reperimento efficiente

A causa dell'indipendenza funzionale tra sottosistemi, le stesse caratteristiche per IU e LA possono essere ottenute con GD diversi...

Per esempio, una tipica applicazione di elaborazione di testi mette a disposizione diversi formati di file per memorizzare i testi

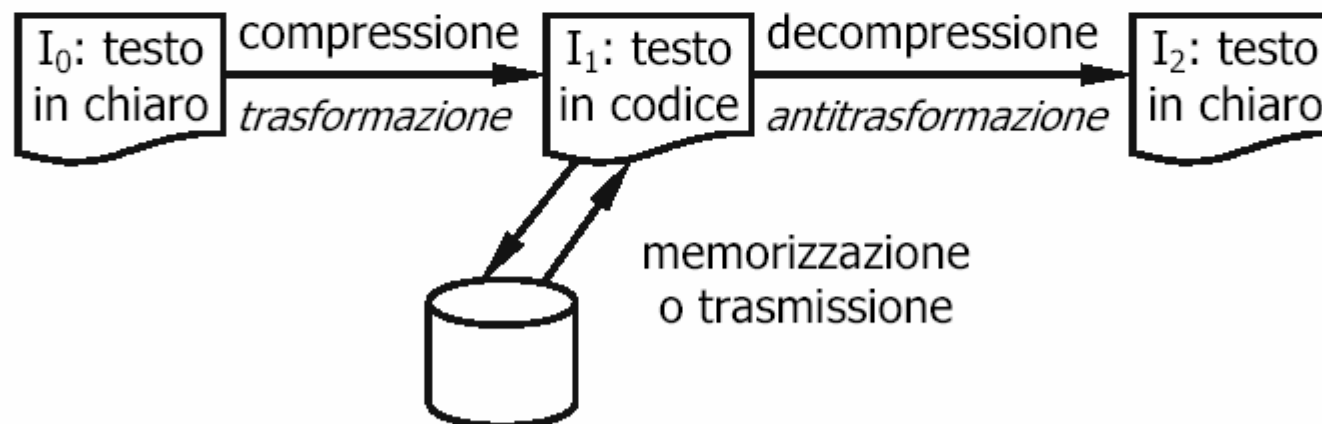
Il sottosistema GD diventa critico quando utenti che usano applicazioni diverse vogliono **scambiarsi dei dati**

I servizi di compressione dei dati

nel caso di:

- memorizzazione su un supporto di capacità scarsa
- trasmissione su un canale di capacità scarsa

si impiegano tipicamente **servizi di compressione dei dati**



PROGRAMMA DELLA LEZIONE

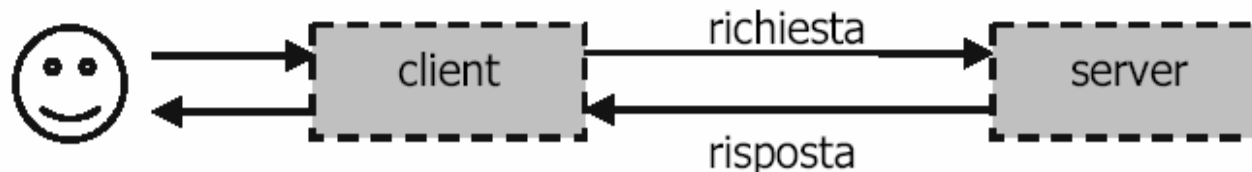
- **Applicazioni come sistemi distribuiti: generalita'**
- **Applicazioni come sistemi distribuiti: esempi: Web, posta elettronica**
- **Ipertesti**

Applicazioni come sistemi distribuiti

I sottosistemi di cui un'applicazione è costituita possono risiedere (ed essere in esecuzione) su calcolatori diversi connessi in rete: si parla in tal caso di **applicazioni distribuite**

L'architettura più diffusa per la realizzazione di applicazioni distribuite è chiamata **client-server**:

- il programma client, con cui opera l'utente, include il solo sottosistema IU; grazie a esso l'utente produce una **richiesta** da inviare al server e ottiene da esso la relativa **risposta**
- il programma server, in esecuzione su un calcolatore raggiungibile sulla rete, include i sottosistemi LA e GD; ricevendo una **richiesta** dalla rete, esso la elabora e produce una **risposta**, che reinvia sulla rete



ESEMPIO SEMPLICE

Applicazione che effettua la somma di due numeri realizzata in architettura client-server

client: accetta l'input (due numeri); lo invia al server attraverso la rete; riceve l'output (la somma) dal server; visualizza l'output ad uso dell'utente

server: riceve l'input (due numeri); effettua la somma (attraverso il sottosistema di Logica Applicativa); invia l'output (la somma) al client

IL WEB COME SISTEMA DISTRIBUITO

**Il World Wide Web (WWW) e' un sistema distribuito
di tipo client - server**

- Client: il browser**
- Server: il server web**

IL WEB COME SISTEMA DISTRIBUITO

Step 1: l'utente si connette a internet e mette in esecuzione il browser; quindi scrive nella barra dell'indirizzo:

<http://www.cbcl.mit.edu/papers/paper.pdf>

e preme 'enter'

IL WEB COME SISTEMA DISTRIBUITO

Step 2: Il browser genera automaticamente un testo del tipo:

GET /papers/paper.pdf HTTP/1.1

User-Agent: Mozilla/4.4 (compatible; Windows XP 2.0)

e lo invia al server www.cbcl.mit.edu come richiesta

IL WEB COME SISTEMA DISTRIBUITO

Step 3: Il server web cerca il file richiesto (paper.pdf) e, trovatolo, produce un file del tipo:

HTTP/1.1 200 OK

Server: Microsoft-IIS/2.0

content-type: pdf

content-length: 14

che invia come risposta al browser

IL WEB COME SISTEMA DISTRIBUITO

Step 4: il browser visualizza il contenuto del file

DUE PROBLEMI

Problema 1: nella comunicazione tra client e server non basta che richiesta e risposta giungano correttamente al destinatario, ma occorre che i due programmi comunicanti ‘parlino la stessa lingua’

Problema 2: per poter accedere ai servizi applicativi messi a disposizione dai server sulla rete e creare quindi un’applicazione distribuita, e’ necessario che ogni servizio sia univocamente identificabile dal client

SOLUZIONE AL PRIMO PROBLEMA

Protocollo applicativo: http (hyper text transfer Protocol) specifica il formato della richiesta del Client (browser) e della risposta del server (web)

HTTP/1.1 200 OK

Server: Microsoft-IIS/2.0

content-type: pdf

content-length: 14

**Intestazione http
con informazioni
sul server e sul
file restituito**

SOLUZIONE DEL SECONDO PROBLEMA

**schema uniforme di identificazione applicativa:
a ogni risorsa presente nella rete vengono
assegnati un metodo di accesso e un indirizzo,
chiamati congiuntamente URL (Uniform Resource
Locator)**

Esempio:

http://profs.sci.univr.it/didattica/calendario.html



The diagram illustrates the components of the URL **http://profs.sci.univr.it/didattica/calendario.html**. Three arrows point from labels below to specific parts of the URL: an arrow from **protocollo** points to **http://**, an arrow from **server** points to **profs.sci.univr.it**, and an arrow from **Nome file** points to **/didattica/calendario.html**.

protocollo **server** **Nome file**

Posta elettronica

- Anche la posta elettronica funziona con lo schema client-server
 - Un client e un server per il mittente
 - Un client e un server per il destinatario
 - ? si evita che mittente e destinatario siano on-line contemporaneamente

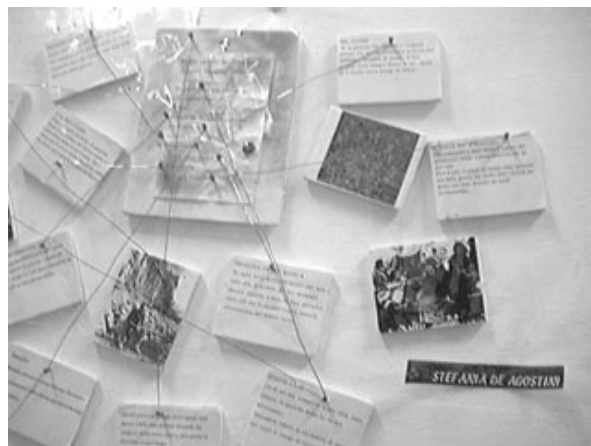
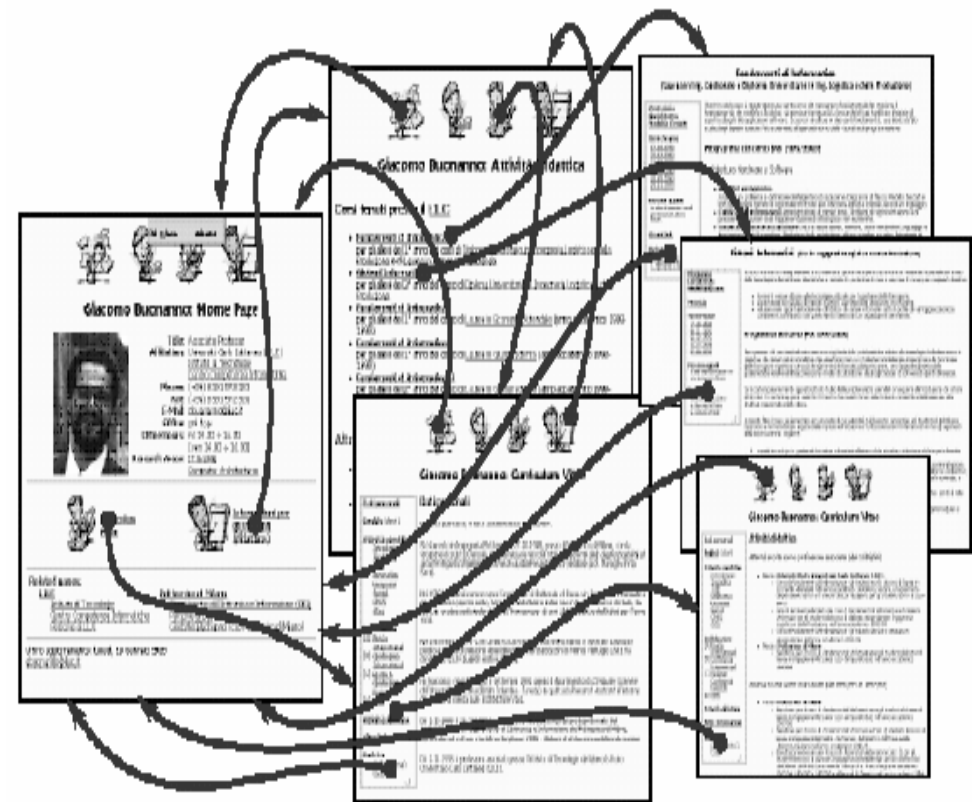
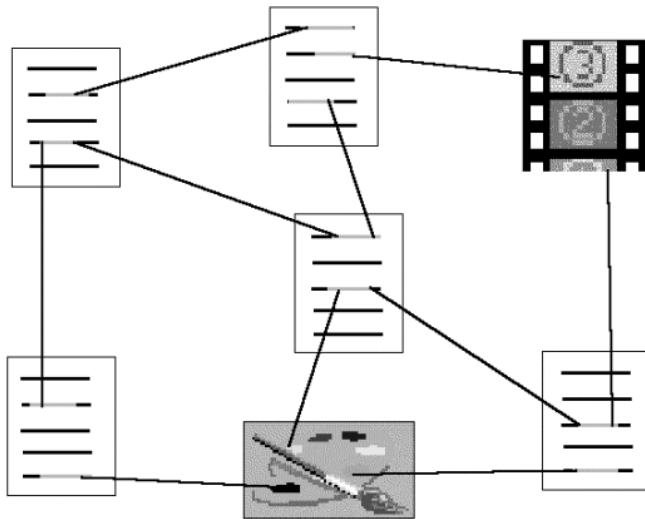
Mittente: SMTP

- Il mittente opera con un client e invia il messaggio al suo server incaricato della spedizione; Simple Mail Transfer Protocol (SMTP)
- Il server del mittente invia il messaggio al server che mantiene la mailbox del destinatario (ancora mediante SMTP)

Destinatario: POP3 o IMAP

- Il destinatario accede alla sua mailbox attraverso un client e ottiene il messaggio in esso contenuto; Post Office Protocol 3 (POP3) oppure Internet Message Access Protocol (IMAP)

Ipertesti



Struttura fisica vs struttura logica

- Limiti del testo cartaceo: l'informazione è organizzata in maniera sequenziale (eventualmente l'indice può indicare dei salti).
- L'organizzazione lineare del supporto fisico non corrisponde ad una organizzazione lineare dei contenuti (es. enciclopedia, vocabolario, etc.).

Struttura fisica ? struttura logica

IPERTESTI

- **Iper testo:** documento la cui struttura di consultazione non è lineare cioè tale che le parti di cui è logicamente costituito sono organizzate in successione ma secondo una struttura complessa

Da Sequenziale a Random (***Interattività***)

IPERTESTI

- Nodi
- Anchor point (punti di aggancio) – link o hyperlink
- Di solito sono attivati da un click del mouse (*point and click*)



Non solo nella
pre-galileiana e
in moto  un
è necessaria un

Metainformazioni

- Le informazioni sull'organizzazione del documento vengono chiamate meta-informazioni.
- I sistemi di gestione degli ipertesti mirano all'automazione delle procedure di trattamento delle meta-informazioni presenti nel documento.

Iper testo multimediale distribuito

- Multimediale
 - Immagini
 - Suoni
 - Video
 - Programmi (script)
- Distribuito
 - Giace su più calcolatori (es: Wolrd Wide Web o WWW)
 - Documenti modulari

Gestione di dati ipermediali

- I sistemi di gestione di ipermedia devono essere in grado di trattare dati diversi (testi, suoni, immagini, etc.).
- E' richiesta l'adozione di uno schema per la formalizzazione del tipo relativo a ciascuna entità presente nel documento.
 - Multipurpose Internet Mail Extensions (MIME), impiegato per abilitare le applicazioni a interpretare correttamente il contenuto di una pagina web e di posta elettronica.

MIME: esempi

- MIME per file html
 - Tipo: text
 - Sottotipo: html
 - Estensioni: .htm, .html
- MIME per file audio
 - Tipo: audio
 - Sottotipo: x-wav
 - Estensioni: .wav

IPERTESTI: HTML

- HyperText Markup Language
- È il linguaggio per creare ipertesti multimediali interattivi e distribuiti.

HyperText Markup Language (HTML)

- E' il formato dei file che realizzano un ipertesto
- Un file HTML contiene sia del testo, che deve essere visualizzato come tale, sia le istruzioni dette *markup* o *tag*
- Ogni tag HTML è identificato da un nome che viene scritto nel file delimitandolo da parentesi angolari:

<nometag>

Tag - esempi

- Questo
è un <P>esempio



Questo

è un

esempio

Inserimento di Immagini

- E' possibile inserire un immagine (memorizzata su un file) in un documento HTML

Questa `` è un'immagine



Che produce: Questa è un'immagine

n.b. l'immagine non fa parte del file HTML!

Formattazione

- Questo <U> è un <I> esempio </I> </U>



- Questo è un *esempio*

n.b. <U> = underline; <I>= italic

Anchor point

- `<H1> Home page </H1>
Michele Piana `



- Home page Michele Piana

RIFERIMENTI AL LIBRO

- **Applicazioni: paragrafi 8.4, 8.4.1, 8.4.2**
- **Ipertesti: paragrafi 3.4, 3.4.1**