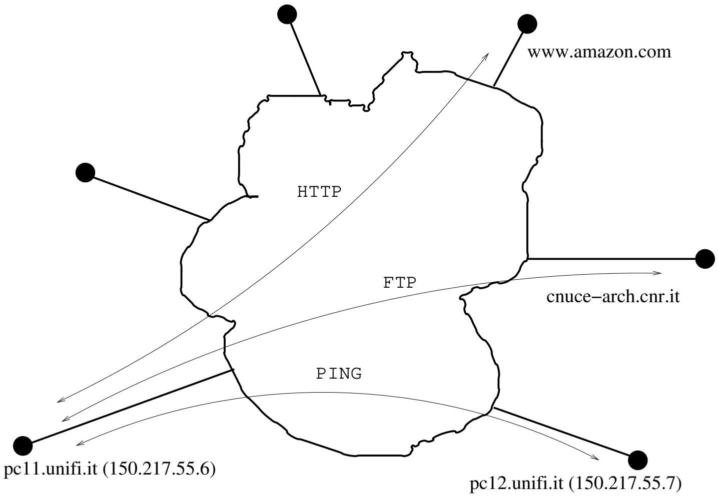
INTRODUZIONE A INTERNET
per studenti universitari
3-1: cos'è e come funziona WWW

Dr. Giorgio F. Signorini
Dipartimento di Chimica
Università di Firenze
signo@chim.unifi.it
http://www.chim.unifi.it/~signo

4 febbraio 2010

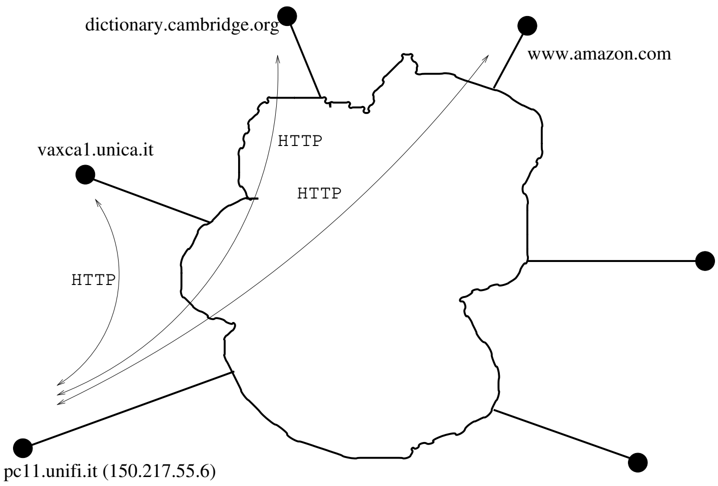
Internet e vari protocolli

- Dal mio host posso contattare diversi host in Internet usando diverse funzioni (o protocolli applicativi): PING, FTP, HTTP, ...



WWW come rete di server HTTP

- Usando HTTP posso contattare solo un insieme di server che formano un *sottospazio* di Internet:



WWW=World Wide Web

- WWW è quella *parte di Internet* che può essere visitata da un client HTTP
- si può anche vedere il WWW come quell'(enorme) *insieme di documenti* interconnessi, distribuiti in tutto il mondo, accessibili via Internet con il protocollo HTTP
- L'accesso è basato su un meccanismo richiesta/risposta (client/server)
    - *Server*: il computer dove stanno i documenti, con un programma per trasmetterli usando il protocollo HTTP
    - *Client*: un computer con un programma per ricevere i documenti usando il protocollo HTTP, e per visualizzarli (il *navigatore* o *"browser"*).

Aspetti principali del WWW

- formato dei documenti (HTML)
- indirizzo dei documenti (URL)
- protocollo di comunicazione (HTTP)

Ipertesti

I documenti presenti su WWW sono generalmente *ipertesti*.

- ipertesto
  è un documento in forma elettronica che contiene rimandi attivi (*“link”*) ad altri documenti.
  una serie di documenti di questo tipo costituisce un testo che può essere letto seguendo vari percorsi.
  Esempio: il manuale on-line di un programma (Word)
  Il testo può essere multimediale.
- ipertesto in rete
  su WWW, un ipertesto situato su un certo server può rimandare a documenti presenti su un server diverso
  Formato di ipertesti (multimediali) usato su WWW: HTML

HTML=”HyperText Markup Language”

- HTML è uno standard stabilito “a posteriori” da un Consorzio di società (W3C, 1994). Ultimo (12/1999): HTML 4.01. Sta venendo sostituito da XHTML (più formale)

  - attributi testo: grassetto, corsivo, vari livelli di intestazioni, font, etc.
  - testo strutturato: testata, corpo, titoli, etc.
  - liste, tabelle
  - figure
  - link
  - moduli (“*form*”)
  - aree indipendenti (*“frame*”)
  - animazioni, musica, spreadsheets, (applets)...

- È un linguaggio di programmazione che viene interpretato dal vostro browser. L’aspetto del documento può variare, a seconda del browser. Un maggiore controllo sull’aspetto del documento si ottiene con le “schede di stile” (*style sheets*) introdotte con HTML 4.

Indirizzi WWW: URI e URL

- Per identificare un documento (o più in generale, una "*risorsa*") su WWW si usa un codice con un formato standard, che viene chiamato in generale *Uniform Resource Identifier (URI)*.
- Un *URL (Uniform Resource Locator)* è un URI che specifica come accedere alla risorsa. È quello che informalmente si chiama *indirizzo WWW.*
- Un URL ha il seguente formato:

`http://`	`www.unifi.it`	`/universita/biblioteche`
1	2	3

1. schema di accesso della risorsa (protocollo)
2. host su cui sta la risorsa
3. percorso per accedere alla risorsa su quell'host

9 / 38

nota: maiuscole in URL

`http://`	`www.unifi.it`	`/universita/biblioteche`
1	2	3

- 1 e 2: <u>non c'è distinzione</u> tra maiuscole e minuscole
- 3: contiene nomi di directory (cartelle) e file, che in generale <u>fanno distinzione</u> tra maiuscole e minuscole
- quindi l'URL precedente si può anche scrivere
  - HTTP://www.unifi.it/universita/biblioteche
  - http://WWW.UNIFI.IT/universita/biblioteche
- ma NON
  - http://www.unifi.it/universita/BIBLIOTECHE

10 / 38

WWW: oltre HTTP

- risorsa=anche immagini, caselle di posta, programmi, ...
- schemi= `http`,`ftp`,`telnet`,`mailto`,`file`,...
  - c'è una corrispondenza (non stretta) tra *schemi* e *protocolli applicativi*
    Es: `ftp://garbo.uwasa.fi/software`
- Perciò si può dare una definizione più generale del WWW:

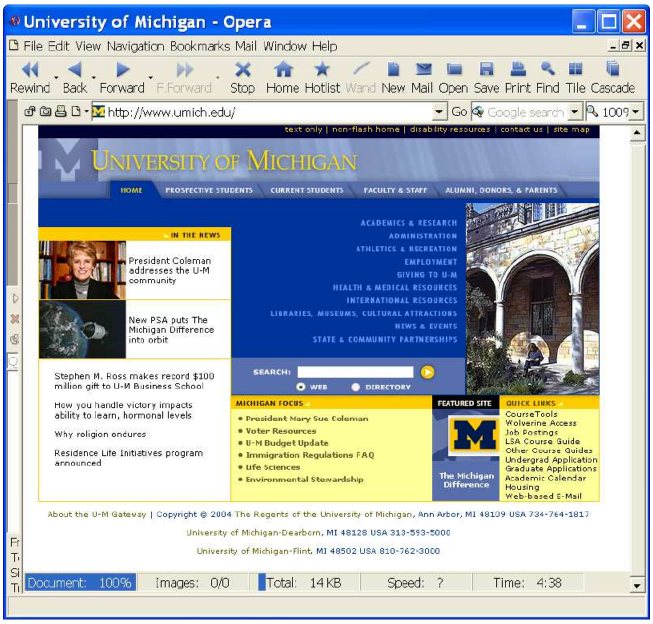
***WWW è un enorme insieme di* risorse *interconnesse, distribuite in tutto il mondo, accessibili via Internet* ed indentificate da un URI**

IL BROWSER

Il browser

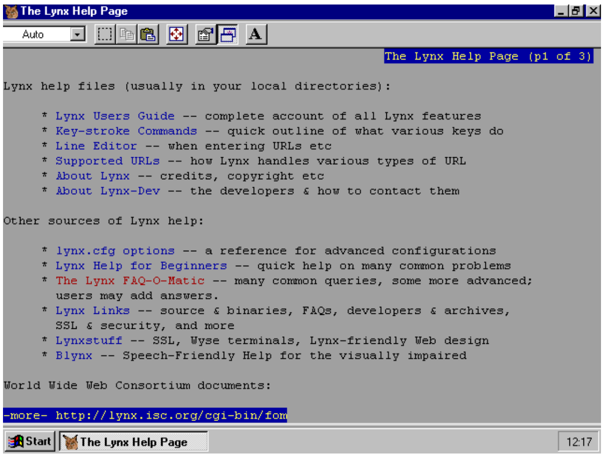
- Il navigatore ("*browser*") è il programma cliente che permette di accedere al WWW, interagendo con i server attraverso HTTP ed altri protocolli come FTP (cfr. gli schemi di un URL)
- p.es.: Netscape, Firefox/Mozilla, Internet Explorer, Opera, Lynx, konqueror, ...

Il browser Opera



Il browser Konqueror

Un browser solo-testo: lynx



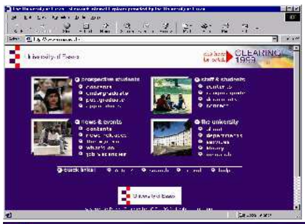
16 / 38

Differenze

- Browser diversi sono equivalenti tra loro, ma non uguali:
  - conformi a livelli HTML diversi
  - riconoscono, o no, elementi HTML non-standard
  - poiché HTML non stabilisce strettamente come deve essere riprodotto un certo elemento (es. un titolo), ma solo la sua funzione nel testo, l'aspetto del documento può variare da un navigatore all'altro
- Spesso contengono altre funzioni, come un programma utente per la posta elettronica (es. Mozilla “suite”)

17 / 38

La stessa pagina su browser differenti



- Internet Explorer 5, WebTV e AvantGo fanno vedere la stessa pagina in modo diverso (vedi questo sito)

18 / 38

Elementi essenziali di un browser

- una riga in cui inserire URL
  - non confonderla con la riga in cui si inseriscono le parole chiave per fare una ricerca!
- una zona dove mostrare la risorsa (con o senza grafica)
- la possibilità di seguire un link (facendo clic o dando invio)

19 / 38

Funzioni utili in un browser WWW

- `cambiare la pagina iniziale di navigazione (7.1.1.4)`[a]
- `salvare una pagina web come file (7.1.1.5)`
- `caricare i testi ma non le immagini (7.1.2.4)`
- `visualizzare le immagini (7.1.2.3)`
- `creare un segnalibro (7.4.1)`
- cercare una parola entro la pagina visualizzata
- ricaricare la pagina
- disattivare la cache
- usare la cronologia per ricaricare una pagina già visitata
- cancellare la cache e la cronologia
- visualizzare il testo HTML
- disattivare le finestre pop-up
- usare le schede (tab)

[a] riferimento alle abilità richieste per la patente europea di guida del computer (ECDL), [cfr. `http://www.aicanet.it/ecdl/ecdl.htm`]

Cosa c'è sotto un link

- Un link è fatto di due elementi:

  1. il testo (spesso evidenziato in modo particolare) su cui fare clic
  2. l'URL da caricare

- Esempio:

```
Per ulteriori informazioni cliccare qui
```

“phishing”

- Questo meccanismo può essere sfruttato per atti di pirateria tendenti a portarti ad un URL diverso da quella che ci si aspetta, ad esempio:

```
Gentile correntista, per favore aggiorni i suoi dati su
sito-sicuro-banca.it
```

- Si può vedere il vero URL a cui rimanda un link
  - in molti browser, nella barra di fondo quando si sfiora il link
  - se si guarda il testo HTML della pagina

22 / 38

IL WWW “ATTIVO”

23 / 38

Eseguire programmi in WWW

- In generale WWW si usa per **trasferire dati** da un server a un client
  - i dati sono già presenti sul server, che deve solo inviarli
  - il client visualizza i dati ricevuti senza eseguire operazioni ulteriori
- In certi casi il trasferimento innesca l’**esecuzione di programmi**
  - sul server: per preparare i dati *ad hoc* prima di trasferirli (creazione di *pagine WWW dinamiche*)
    - in questo modo si evita di mantenere sul server troppi dati già confezionati
  - sul client: per elaborare i dati dopo il trasferimento
    - si evita di trasferire dati troppo grossi: li si ricostruisce all’utilizzo

24 / 38

(A) sul cliente (JavaScript, Java, ...)

1. La pagina HTML può contenere istruzioni in un linguaggio interpretato dal browser (JavaScript o VBScript)
   - questi linguaggi consistono essenzialmente in funzioni assenti da HTML, che permettono ad es. di scrivere la data di oggi, aprire una nuova finestra, etc.
2. Oppure può contenere un riquadro in cui eseguire un programma semi-compilato ("*applet*"), scritto in linguaggio Java.
   - Caratteristiche di Java:
     - è indipendente dalla piattaforma (hw e sw) cliente
     - l'aspetto della grafica è lo stesso su tutti i browser (perciò viene utilizzato essenzialmente per la grafica)
     - contiene funzioni che sfruttano la rete

25 / 38

Nota: JavaScript, Java etc. sono pericolosi?

- Le istruzioni JavaScript, VBS, ... o gli *applet* Java vengono eseguiti sul **cliente** (lo scambio con il server è solo il trasferimento del testo del programma)
- perciò JavaScript, Java, etc. sono progettati per essere il più possibile innocui verso il computer (non possono scrivere su disco)
- però è dimostrato che è possibile usarli maliziosamente per danneggiare un PC (ad es. bloccarlo)
- quindi molti esperti di sicurezza **consigliano di disattivare JavaScript e Java** sul proprio browser, attivandoli soltanto quando si carica una pagina sicura

26 / 38

(B) sul server (CGI, PHP, ASP, ...)

- Funzionamento: certe risorse sul server, invece di essere inviate direttamente al cliente, prima eseguono un programma, il cui risultato è un documento HTML che viene trasmesso al cliente.
- Ad esempio: la ricerca su un database:
  `http://opac.unifi.it/opac/short?author=alighieri`
  - Tipicamente, l'URL contiene un "?" dopo il quale vengono i parametri da passare al programma del server.
- I linguaggi di programmazione più diffusi sono:
  - CGI
  - (più recente) PHP
  - (Microsoft) ASP

27 / 38

Content Management System

- Un esempio di esecuzione di programmi sul lato server è il "sistema di gestione dei contenuti" (Content Management System, CMS)
- Per capirne la logica, pensiamo ad un esempio: il sito WWW di un giornale.
  - i giornalisti scrivono pezzi su vari argomenti e li archiviano in un database con qualche parametro (data, argomento, etc)
  - il redattore prepara varie edizioni on-line del giornale selezionando i pezzi che rispondono a certi parametri
  - lo stesso può fare un visitatore: ad esempio, prepararsi un'edizione personalizzata che contenga soltanto le notizie di sport
- Se un server adotta il sistema CMS, in generale le pagine vengono preparate "al volo" su richiesta del cliente

28 / 38

CMS: vantaggi e svantaggi

- vantaggi:
  - si separa il contenuto (che può essere immesso da diversi autori) dalla forma (che è gestita in modo unificato)
- svantaggi:
  - non possono essere ricercate con un URL
  - non sono visibili da tutti, ad es. da un motore di ricerca (v. avanti)

29 / 38

MEMORIE

30 / 38

HTTP è “smemorato”

- In HTTP (a differenza di altri protocolli applicativi, come FTP), le connessioni vengono generalmente chiuse una volta che una richiesta è stata soddisfatta.
- Il server si “dimentica” della precedente transazione.
- Questo è ideale per il WWW, in cui le pagine molto spesso contengono dei link a pagine ospitate da altri server: è meglio “liberare” un server appena ci ha fornito una pagina.
- A volte però è utile conservare memoria delle transazioni precedenti.
- Lo si fa dal lato del cliente. Ci sono tipi due meccanismi:
  - cache/proxy: copia di pagine già visitate
  - cookie: dati riassuntivi su una particolare transazione

31 / 38

Cache del browser

- La “cache” è una memoria del vostro computer su cui sono registrate le ultime pagine che avete aperto con quel browser
- serve per non doverle ricaricare tutte le volte, magari da un sito poco raggiungibile
- ci sono due tipi di cache:
  - cache di memoria: cancellata quando si chiude il browser
  - cache su disco: riutilizzata anche nelle successive sessioni del browser
- attenzione! dopo un po’ è bene che le pagine “scadano”, perché l’informazione potrebbe non essere più aggiornata (es. la homepage di un quotidiano)

32 / 38

Proxy

- Un proxy HTTP è un server esterno che tiene sul suo disco la cache vostra e di altri utenti.
- Serve quando la stessa pagina è visitata spesso da un gruppo di utenti diversi
- Si può configurare questo servizio in modo tale che alcune pagine (che cambiano spesso) NON siano accedute tramite proxy, ma direttamente

33 / 38

Cookie

- I “cookie” sono dati sintetici inviati da un server HTTP ad un browser che lo sta consultando, per essere salvati sul disco del cliente e letti dal server in una successiva connessione
- Un cookie è per esempio un modo per:
  - salvare preferenze, password, etc
  - mettere nel “carrello della spesa” gli oggetti che si sono comprati prima di pagarli tutti insieme
- Preoccupati della riservatezza? Ricordate che un cookie non contiene su di voi più informazione di quella che comunque avete già dato al server!

34 / 38

GLOSSARIO

35 / 38

Glossario di termini comuni

- “Pagina”=un singolo documento, individuato da un URL
  - Poiché le pagine sono interconnesse, ad una stessa pagina si può arrivare da vie differenti.
  - Una pagina può essere sia punto di arrivo che punto di partenza.
  - Alcune pagine sono più “periferiche” (es. capitoli di un libro), altre sono più “di passaggio” (es. elenco di biblioteche)
- “sito”: un server, o la sua pagina principale, o l’insieme dei documenti presenti su un server, o una parte di questo insieme che costituisce una struttura omogenea
- *“Home page”*: pagina che costituisce un punto di partenza specifico (di una persona, un’organizzazione, un argomento).
- “Portale”: punto di partenza generico. Un ISP ha sempre un portale che contiene come minimo un catalogo per soggetto e un motore di ricerca

36 / 38

Bibliografia: WWW

- D. E. Comer, Internet e reti di calcolatori, Pearson Education Italia, Collana Accademica, 2003
- Ed Krol, The Whole Internet user's guide and catalog. O'Reilly and Associates, 1996
- Ed Krol, Internet, Milano, Jackson libri, 1994.
- Wikipedia (Enciclopedia online), per definizioni dei termini: `http://www.wikipedia.com/wiki/Main+Page`
- R. Fielding et al.: Hypertext Transfer Protocol – HTTP/1.1.. RFC 2616 `http://www.ietf.org/rfc/rfc2616.txt`
- W3C (World Wide Web Consortium): HTML 4.01 Specification `http://www.w3.org/TR/REC-html40/`
- T. Berners-Lee: Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396 `http://www.ietf.org/rfc/rfc2396.txt`