



Internet tehnologije

World Wide Web



Uvod

- Hipertekst – velika prednost je jednostavno uspostavljanje i predstavljanje relacija između dokumenata - linkovi
- Izvorna ideja 1945.
- Pojam “hipertekst” – 1960. – Xanadu sistem
- World Wide Web – CERN 1989. – Tim Berners-Lee predložio kreiranje mreže elektronski povezanih dokumenata, kreirao prvu verziju HTTP protokola 1990. godine, učestvovao u razvoju URI i HTML
- Web server / web client
- Danas je primjetan prelaz sa pretežno tekstualnih dokumenata na multimedijalnu predstavu informacija
- Web postaje medij za dvosmjernu komunikaciju

Osnovni elementi WWW



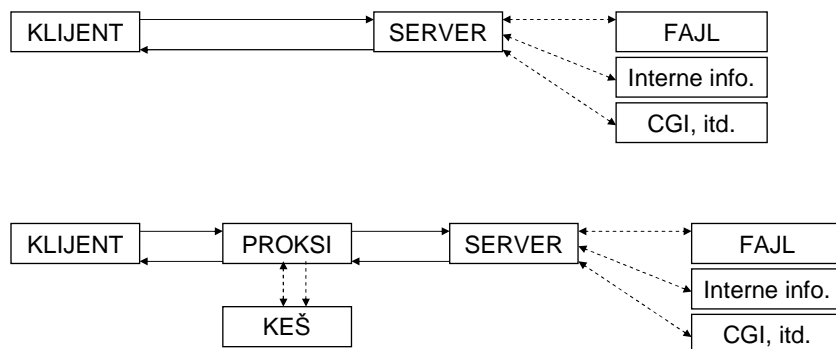
- Generalno posmatrano postoje tri najvažnija elementa: HTML, HTTP i URI
- HyperText Markup Language – HTML
 - Tekstualni jezik za opis izgleda dokumenta
 - Počiva na ideji upotrebe “tagova” za uspostavljanje veza između dokumenata i formatiranje prikaza dokumenta
 - Izuzetno popularan – napravljen je veliki broj drugih jezika slične namjene

Elementi WWW (2)



- Hypertext Transfer Protocol – HTTP
 - Klijent – server protokol baziran na TCP/IP
 - Protokol za prenos dokumenata
 - U novijim implementacijama je funkcionalnost protokola značajno proširena (multimedija, striming, keširanje, ...)
- Uniform Resource Identifiers – URI
 - Identifikuju resurse dostupne putem Interneta sa ciljem lakšeg povezivanja

Funkcionisanje WWW



** DNS komunikacija nije prikazana*

HTML

- Zasnovan na Standard Generalized Markup Language (SGML) jeziku
- Osim teksta sadrži i informacije potrebne za pravilan prikaz dokumenta
- Tagovi
 - `<element par1="vr1" par2="vr2" ...>`
 - `<TITLE>Naslov dokumenta</TITLE>`
 - `Ovo <l>je</l> proba`

HTML primjer

```
<HTML>
<HEAD>
<TITLE>Mali primjer</TITLE>
</HEAD>
<BODY>
<H1>Naslov</H1>
Prva linija<BR>
Druga linija<BR>
<B>Bold<I>bold italic</I></B><HR>
<A HREF="http://www.etfbl.net">Link na ETF</A>
</BODY>
</HTML>
```

URI, URL, URN

- Uniform Resource Identifier (URI)
 - Uniform Resource Locator (URL)
 - Uniform Resource Name (URN)
- Danas se gotovo isključivo koriste URL-ovi jer su URN standardi još u fazi razvoja
- Vrste URL-ova
 - Apsolutni
 - Relativni



Anatomija URL-a

- Opšti oblik

`scheme://user:password@host:port/path
;params?query#fragment`

- Za HTTP ima oblik

`http://user:pass@host:port/path?query#bkmark`

- Obično se ne koriste svi elementi

`http://host/path`



Tolerancija

- Zbog jednostavnosti upotrebe WWW-a tolerišu se i nepotpuni zahtjevi
 - “www.etfbl.net” uneseno u web browseru
 - se prevodi u “http://www.etfbl.net/” i takav zahtjev se šalje web serveru
 - server prima takav zahtjev i umjesto da izlista fajlove u osnovnom direktorijumu vraća “http://www.etfbl.net/index.php” (ili index.html, default.htm, itd) – podrazumijevani dokument

HTTP verzije



- U upotrebi su dvije verzije
 - HTTP/1.0 – RFC 1945
 - HTTP/1.1 – RFC 2068, RFC 2616
- Prednosti HTTP/1.1
 - Virtuelni hostovi – više domena na jednoj IP adresi
 - Perzistentne konekcije – klijent može poslati više zahtjeva u jednoj TCP sesiji
 - Parcijalni transfer – prenos samo dijela resursa
 - Poboljšano keširanje i upotreba proksija
 - Poboljšanje bezbjednosti – metodi autentifikacije
 - Klijent i server mogu dogovoriti vrstu ili verziju sadržaja za prenos

HTTP konekcije



- Dvije vrste: *Transitory*, *Persistent*
- Transitory
 - HTTP/0.9 i HTTP/1.0
 - Način rada: uspostavlja se veza klijent-server, klijent šalje zahtjev, server šalje odgovor, veza se prekida
 - Prednost je velika jednostavnost, mana je veliki broj zahtjeva od strane klijenta
 - Svaki zahtjev je nova TCP sesija
 - Ovakav pristup je posljedica ranog izgleda weba kada su dokumenti bili gotovo isključivo tekstualni i sadržani u jednom fajlu

HTTP konekcije (2)



- Persistent
 - HTTP/1.1
 - TCP konekcija se ne zatvara automatski već se može koristiti i za naredne zahtjeve
 - Prednost je smanjivanje broja konekcija a samim tim i manja potrošnja procesorskih, memorijskih i mrežnih resursa
 - **Pipelining** – mogućnost da klijent traži veći broj npr. fajlova u jednom zahtjevu
 - Mana je znatno kompleksniji način prenosa podataka naročito u slučaju pipelininga
 - Persistent je podrazumijevani način rada u HTTP/1.1 iako je ostavljena mogućnost da klijent zahtijeva eksplicitno zatvaranje konekcije nakon svakog zahtjeva/odgovora

HTTP poruke



- Klijent i server razmjenjuju HTTP poruke
- Poruke su tekstualne i bazirane su (približno) na RFC (2)822 i MIME standardima
- Opšti format HTTP poruke je

`<početna linija>`
`<zaglavlje poruke>`
`<prazna linija CR/LF>`
`[tijelo poruke]`
`[nastavci poruke]`



HTTP poruke

- Početna linija je tekstualna linija koja obilježava poruku kao zahtjev ili odgovor
 - Zahtjev sadrži metod i URI traženog resursa
 - Odgovor sadrži statusne informacije
- Zaglavlje je tekstualnog tipa, sadrži parove
 - ime-zaglavlja: vrijednost
- Tijelo poruke je opciono i ne mora biti tekstualno



Format HTTP zahtjeva

- Početna linija – *request line*
`<METHOD> <request-uri> <HTTP-version>`
- METHOD – GET, POST, HEAD, ...
- Request-uri – resurs koji nam treba
- HTTP-version – korištena verzija HTTP-a
- Zaglavlja su opciona osim za HTTP/1.1 u kome je *Host: ime_hosta* obavezno
 - Postoje tri vrste: *General*, *Request* i *Entity*

Primjer HTTP zahtjeva



`GET /index.php HTTP/1.1` Statusna linija

`Connection: close` General headers

`Host: www.etfbl.net` Request headers

`Accept: text/plain, text/html`

`User-Agent: MojBrowser 1.0`

`<prazna linija CRLF>`

Format HTTP odgovora



- Početna linija – *status line*
`<HTTP-version> <status-code> <reason>`
- Status-code je trocifren i sličan je kao kod drugih protokola (SMTP, FTP, itd)
- Reason je tekstualno objašnjenje statusnog koda
- Zaglavlja su slična kao i kod zahtjeva
 - Postoje tri vrste: *General*, *Response* i *Entity*

Primjer HTTP odgovora

HTTP/1.1 200 OK	Statusna linija
Connection: close	General headers
Server: Apache/1.3.33	Response headers
Accept-Ranges: bytes	
Content-Type: text/html	Entity headers
Content-length: 2140	
<prazna linija CRLF>	
<HTML>	Message body
...	
</HTML>	

HTTP metodi

- Uobičajeni metodi
 - **GET** – server treba da u odgovoru pošalje traženi resurs
 - **HEAD** – slično kao GET ali se vraćaju samo zaglavlja bez tijela poruke
 - **POST** – šalje podatke od klijenta prema serveru na dalju obradu
- Ostali metodi: PUT, DELETE, OPTIONS, TRACE, itd

Primjer statistike HTTP metoda

method	request	%	Byte	%
GET	43029	95.38	287562K	96.91
CONNECT	820	1.82	6750K	2.28
POST	1170	2.59	2342K	0.79
OPTIONS	34	0.08	34K	0.01
HEAD	48	0.11	16K	0.01
PROPFIND	10	0.02	8K	0.00
NONE	4	0.01	6K	0.00

Statusni kodovi

- Kodovi se mogu podijeliti prema prvoj cifri
 - **1xx** Informational Message
 - **2xx** Success
 - **3xx** Redirection
 - **4xx** Client Error
 - **5xx** Server Error
- Kodovi **x00** se smatraju podrazumijevanim (ako klijent ne zna šta je kod 512 posmatra ga kao 500)



Najčešći kodovi

- 200 OK
- 301 Moved Permanently
- 304 Not Modified
- 400 Bad Request
- 401 Unauthorized
- 403 Forbidden
- 404 Not Found
- 405 Method Not Allowed
- 406 Not Acceptable
- 500 Internal Server Error
- 501 Not Implemented



Cookies

- Problem sa HTTP je u tome što je on **stateless** protokol – ne pamti stanje između dva zahtjeva istog klijenta (čak i u HTTP/1.1 persistent načinu rada) i posmatra ih kao odvojene događaje
- RFC 2965 - HTTP State Management Mechanism
- Cookie – fajl koji se smješta na klijentsku stranu i logički povezuje transakcije, sadrži određene informacije potrebne za to (username/password, ID sesije, itd.)
- Postoje brojni problemi: prenos povjerljivih informacija, neovlašteno praćenje aktivnosti korisnika, ubacivanje kukija sa drugih servera, upotreba u okviru *spyware* programa, itd.



Bezbjednost

- HTTP se često koristi za prenos osjetljivih informacija
- Autentifikacija – RFC 2617
 - **Basic Authentication** – klasičan pristup username/password autentifikacije (problem u slanju username/passwd u izvornom obliku)
 - **Digest Authentication** – u toku autentifikacije se ne prenose username/passwd u izvornom obliku već se razmjenjuju *digesti* (jednosmjerno šifrovanje)



Bezbjednost (2)

- Problematičan prenos povjerljivih informacija
- Problem sa keširanjem povjerljivih informacija
- Problemi sa kukijima
- Problem sa zahtjevima tipa
`GET http://www.etfb1.net/a.php?user=deda&pass=mraz`
 - Rješenje je upotreba POST metoda
- Ovakvi problemi se obično rješavaju upotrebom HTTPS protokola koji je kombinacija HTTP i SSL protokola (Secure Sockets Layer – SSL)