



Diplomski studij

**Informacijska i
komunikacijska tehnologija:**

Telekomunikacije i informatika

Računarstvo:

Programsko inženjerstvo i
informacijski sustavi

Računarska znanost

Ak.g. 2008./2009.

Raspodijeljeni sustavi

5.

World Wide Web

2.10.2008.

- ◆ osnove World Wide Weba
- ◆ nastanak World Wide Weba
- ◆ izvedba usluge u mreži i programska poptora
- ◆ pojam *Uniform Resource Identifier* (**URI**)
- ◆ zapis sadržaja na Webu (**HTML**)
- ◆ protokol *Hypertext Transfer Protocol* (**HTTP**)

Pripremljeno na temelju predavanja:

Doc.dr.sc. Gordan Gledec, "Internetske usluge: World Wide Web – terminologija, formati i protokoli", FER, Telematičke usluge, ak.g. 2007./8.

- ◆ usluga: globalni hipermedijski informacijski sustav
 - informacijski prostor weba čine *informacijski izvori* ili *sredstva* - *resursi* (engl. *resource*) međusobno povezani hipervezama (engl. *hyperlink*)
- ◆ model izvedbe usluge: klijent-poslužitelj
- ◆ osnovne komponente:
 - zapis izvora: **HTML /XHTML**, jednostavan, prenosiv zapis teksta, mogućnost umetanja hiperveza, korištenje datoteka s drugim medijima (slike, audio, video) u izvornom obliku
 - adresiranje - identifikacija izvora: **URI**
 - način povezivanja i komunikacije: **HTTP**, internetski aplikacijski protokol

- ◆ hipertekst (engl. *hypertext*)
- ◆ hipermedij (engl. *hypermedia*)
- ◆ hiperveza, hiperpoveznica (engl. *hyperlink*)
- ◆ poveznica (engl. *link*)
- ◆ informacijski izvor ili resurs (engl. *resource*)
- ◆ informacijski prostor

Podsjetimo se:

“Komunikacijske mreže”

“Sjednički sloj, prezentacijski sloj, aplikacijski sloj. Usluge i protokoli aplikacijskog sloja u Internetu.” (www.fer.hr/predmeti/kommre)

Pojam informacijskog izvora ili resursa



- ◆ pojam hipermedijskog dokumenta proširuje se pojmom *informacijskog izvora* ili *resursa*
 - u općenitom smislu, “bilo što” što daje informaciju i što se može identificirati

- ◆ obično promatramo konkretne, automatizirane, mrežno dohvatljive informacijske izvore, npr.:
 - elektronički dokument,
 - slika,
 - izvor informacije jasne namjene (npr. tečaj HNB),
 - usluga (HTTP-SMS prilaz),
 - kolekcija resursa.

- ◆ primjer *izvora*: elektronički dokument (“datoteka”)
 - *informacija* koju datoteka pruža je njen *sadržaj* (može biti statički ili promjenjiv)
 - prikaz, odnosno *reprezentacija* informacije se često naziva “Web stranicom”

◆ “klasične” Internetske usluge

- tematski web portali
- pristup datotekama
- *e-mail, webmail, archive mailing lista*
- *news, forumi, blogovi*
- ...

◆ ostale usluge

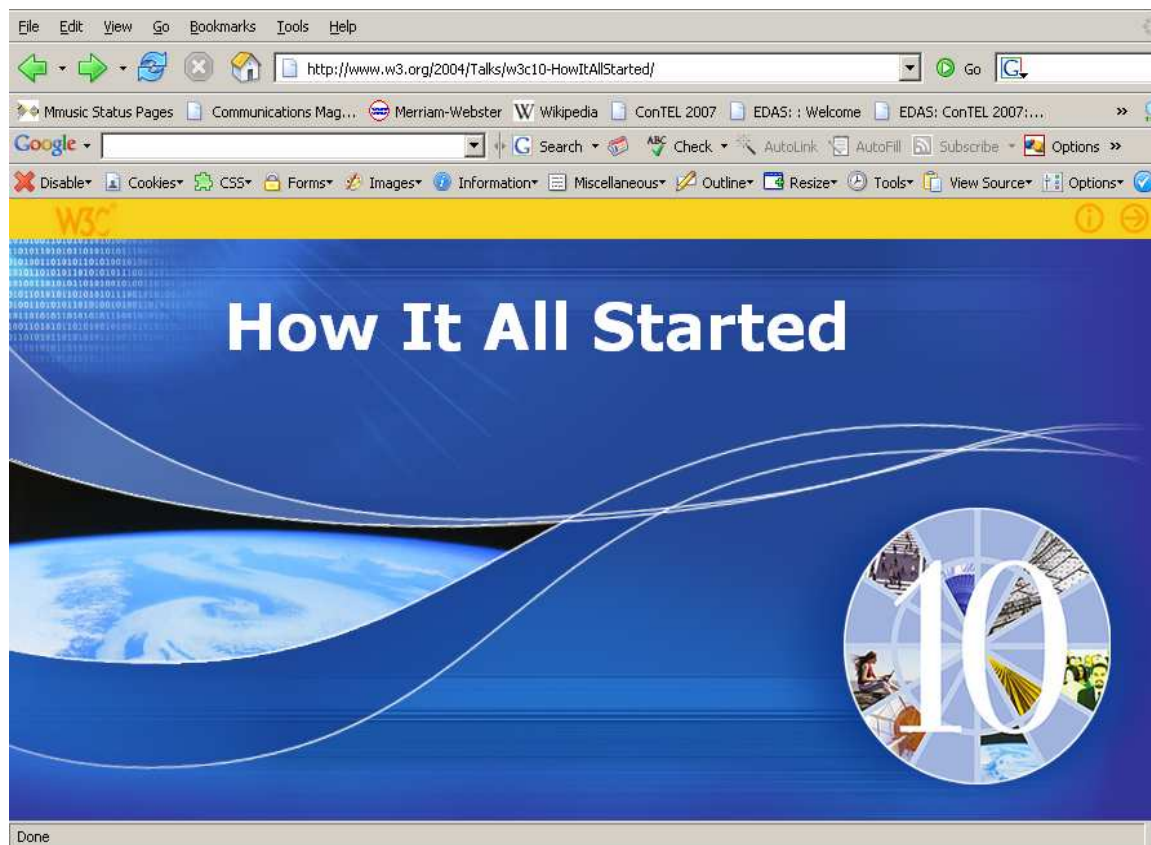
- rezervacijski sustav (avio-karte, hoteli, ...)
- digitalna knjižnica, *on-line* publikacije
- osobno bankarstvo
- studentska služba (upis, prijava ispita, ...)
- ...

Kako je sve započelo ..., riječima izumitelja Web



- ◆ Tim Berners Lee – prezentacija na 10. godišnjici WWW-a

Izvor: <http://www.w3.org/2004/Talks/w3c10-HowItAllStarted/>



Tim Berners-Lee,
W3C Director and
inventor of the World
Wide Web

Počeci World Wide Weba



- ◆ priznati autor ideje - **Vannevar Bush**, 1945.g.
 - **memex** – uređaj koji će proširiti ljudsku sposobnost pamćenja i rukovanja informacijama kroz poveznice među dokumentima (“As We May Think”, The Atlantic Monthly, July 1945)

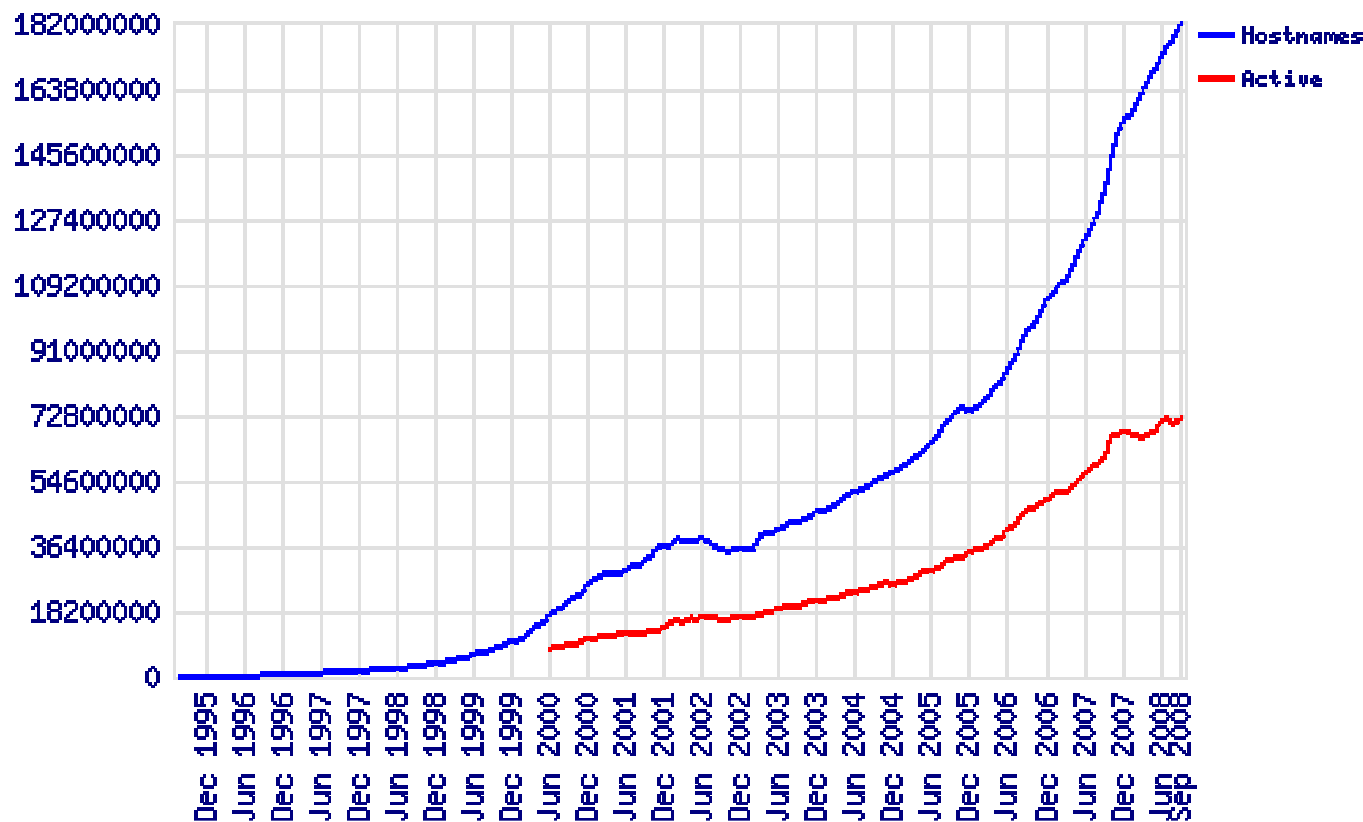
- ◆ počeci realizacije današnjeg WWW-a: CERN  European Organization for Nuclear Research
 - zamisao: omogućiti znanstvenicima iz raznih država jednostavan pristup raznim dokumentima vezanim za projekte
 - prvi prijedlog sustava povezanog hipervezama - **Tim Berners-Lee**, 1989. g.

- ◆ prva javna demonstracija sustava (s tekstualnim sučeljem) 1991. g. na konferenciji *Hypertext '91*
- ◆ prvi preglednik s grafičkim sučeljem, Mosaic, u veljači 1993. g.
- ◆ u Hrvatskoj: počeci 1993. g., Croatian Home Page (kasnije, www.hr) 1994. g. (<http://web.archive.org/web/19961106021324/http://tjev.tel.fer.hr/>)

Trend rasta broja Web-sjedišta (1995-2008)



Total Sites Across All Domains August 1995 - September 2008



“In the **September 2008** survey we received responses from **181,277,835** sites. Of the 4.5 million sites that have been gained this month, more than three-quarters are using Apache.”

Izvor: http://news.netcraft.com/archives/web_server_survey.html

Elements of the Web's Next Generation



Source: <http://web2.wsj2.com>

Tim Berners-Lee: "...Web 2.0 is of course a piece of jargon, nobody even knows what it means. If Web 2.0 for you is blogs and wikis, then that is people to people. But that was what the Web was supposed to be all along..."

Izvor: <http://www-128.ibm.com/developerworks/podcast/dwi/cm-int082206.txt>

Semantički Web

- ◆ *"I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A 'Semantic Web', which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The 'intelligent agents' people have touted for ages will finally materialize."*

– Tim Berners-Lee, 1999

Preporučena literatura:

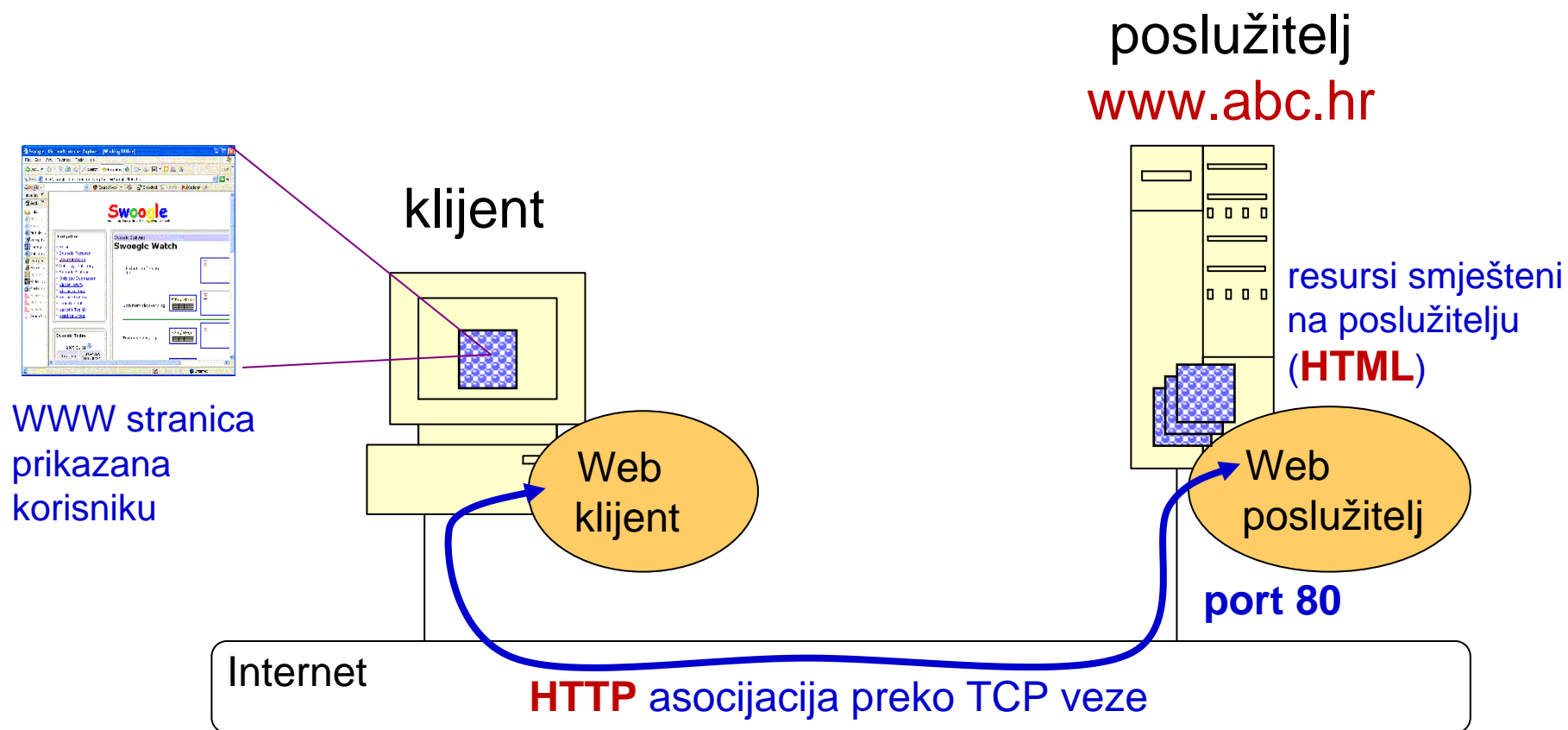
- ◆ **W3C Semantic Web Activity** [Izvor: <http://www.w3.org/2001/sw/>]
- ◆ T. Berners-Lee, Tim; J. Hendler , O.Lassila (May 17, 2001). "The Semantic Web". Scientific American Magazine. [Izvor: <http://www.sciam.com/article.cfm?id=the-semantic-web>]
- ◆ T. Berners-Lee, intervju 7.2.2008.
[Izvor: http://talispodcasts.s3.amazonaws.com/twt20080207_TimBL.html]

Izvedba usluge WWW u mreži



- ◆ model klijent-poslužitelj
- ◆ resurs identificiran putem **URI**

Sustav za upravljanje sadržajem
(CMS – *Content Management System*)



◆ Web **klijent**

- korisnički Web klijent – **preglednik** (engl. *browser*)
 - grafičko ili tekstualno korisničko sučelje za prikaz Web stranice i navigaciju; novije verzije donose više mogućnosti
 - popularni preglednici: Netscape, Mozilla, Firefox, Internet Explorer, Opera, Lynx, ... (uglavnom besplatni)
- automatizirani Web klijent – **robot** ili pauk (engl. *spider, crawler*)
 - program koji samostalno pretražuje Web (ili neki njegov dio) radi prikupljanja podataka, npr. za tražilice

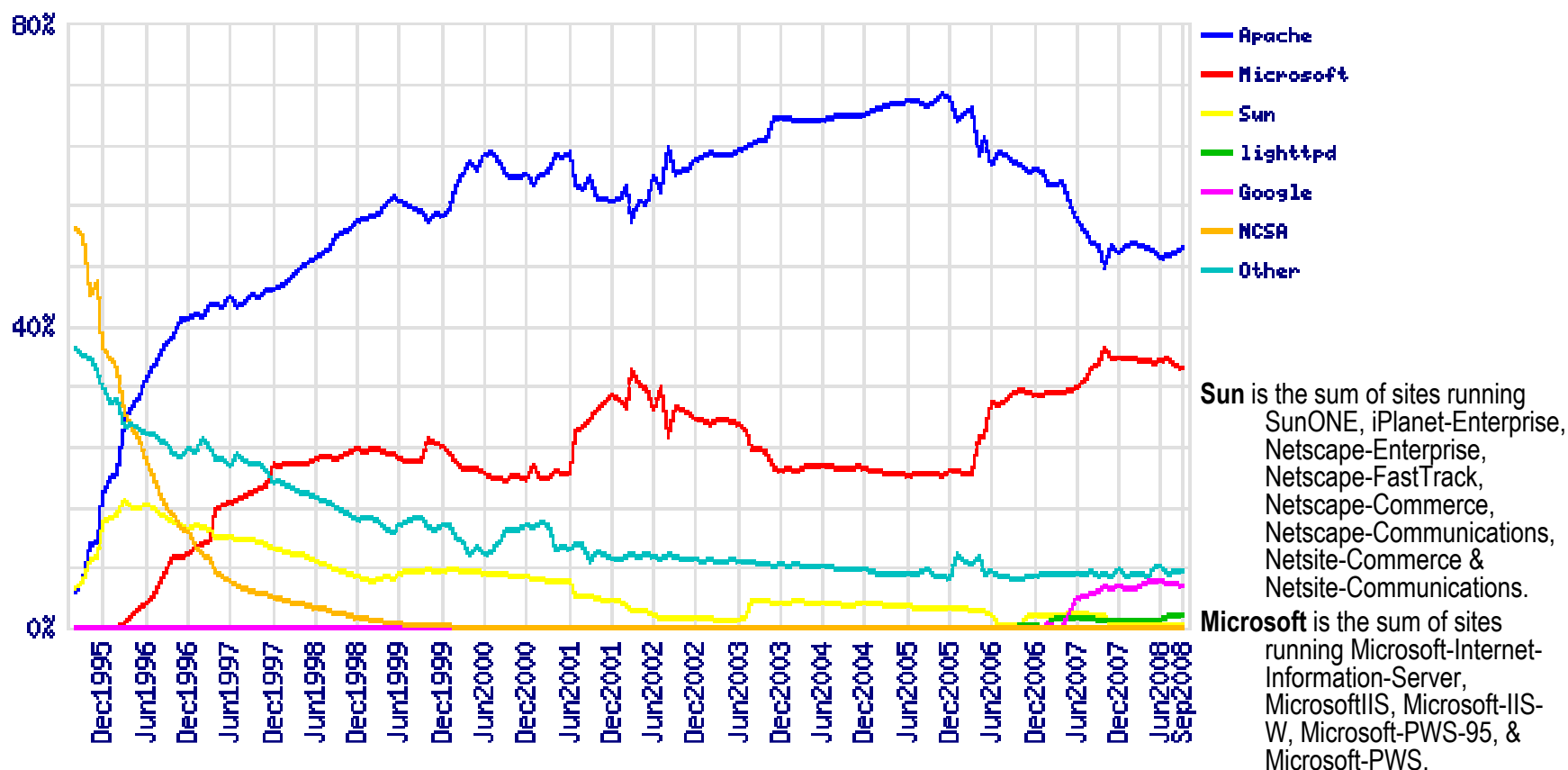
◆ Web **poslužitelj**

- popularni HTTP poslužitelji: Apache HTTP server (besplatan), Microsoft Internet Information Server
- dodatni aplikacijski poslužitelji

WWW poslužitelji (podaci 9/2008)



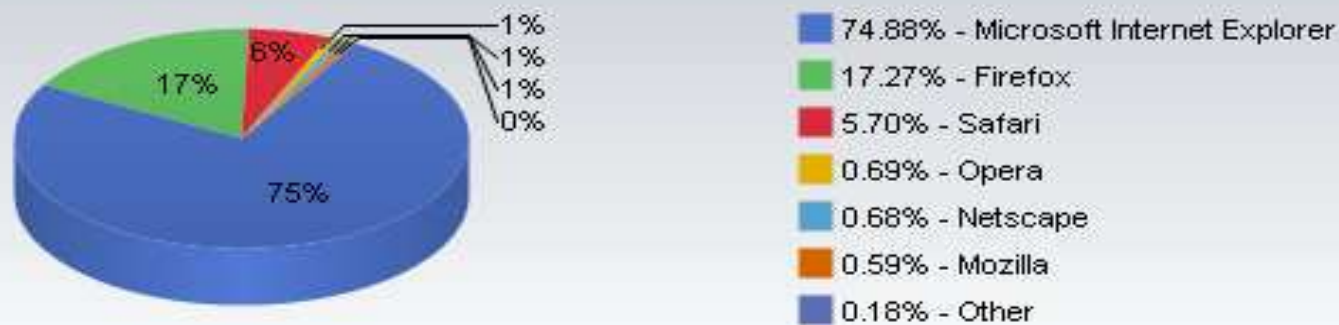
Market Share for Top Servers Across All Domains August 1995 - September 2008



WWW klijenti - *preglednici* (podaci 3-9/2008)




3/2008



Izvor: <http://marketshare.hitslink.com/report.aspx?qprid=0>

9/2008

- 71,52% Microsoft Internet Explorer
- 19,46% Firefox
- 6,65% Safari

- identifikacija izvora: **URI** 
- zapis izvora: HTML (XHTML)
- način povezivanja i komunikacije: HTTP

URI – **U**niform **R**esource **I**dentifier

(uniformni identifikator resursa)

- ◆ **uniformni**: jednoobrazni način zapisa – propisan je oblik
- ◆ **identifikator**: sadrži informaciju nužnu za razlikovanje identificiranog resursa od svih ostalih (\neq identitet!)
- ◆ **resurs**: informacijski izvor; “*bilo što*” što se može identificirati URI-jem

*Pojam URI-ja je središnji pojam u arhitekturi World-Wide Weba.
World Wide Web Consortium (W3C) definira WWW kao “informacijski
prostor u kojem su predmeti od interesa identificirani URI-jima”.*

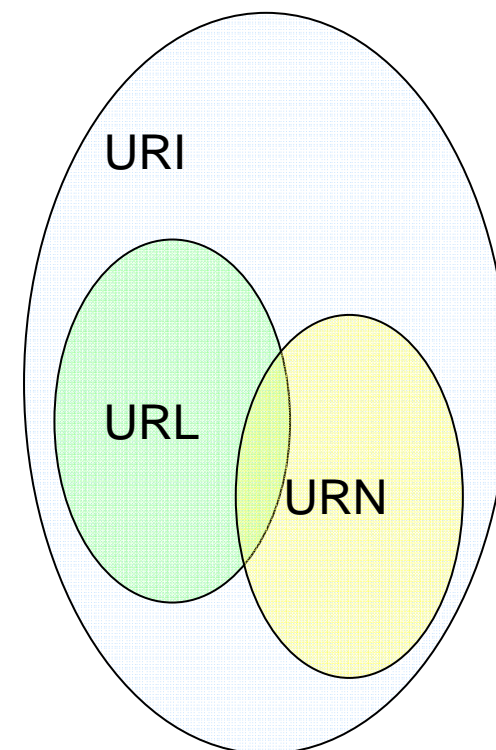
◆ URL – *Uniform Resource Locator*

- resurs se identificira preko svoje (mrežne) **lokacije**
- npr. `http://161.53.19.1:8080`

◆ URN – *Uniform Resource Name*

- stroži zahtjevi na trajnost: moraju se održavati čak i kad resurs koje identificiraju nestane
- npr. `urn:ietf:rfc:2396`

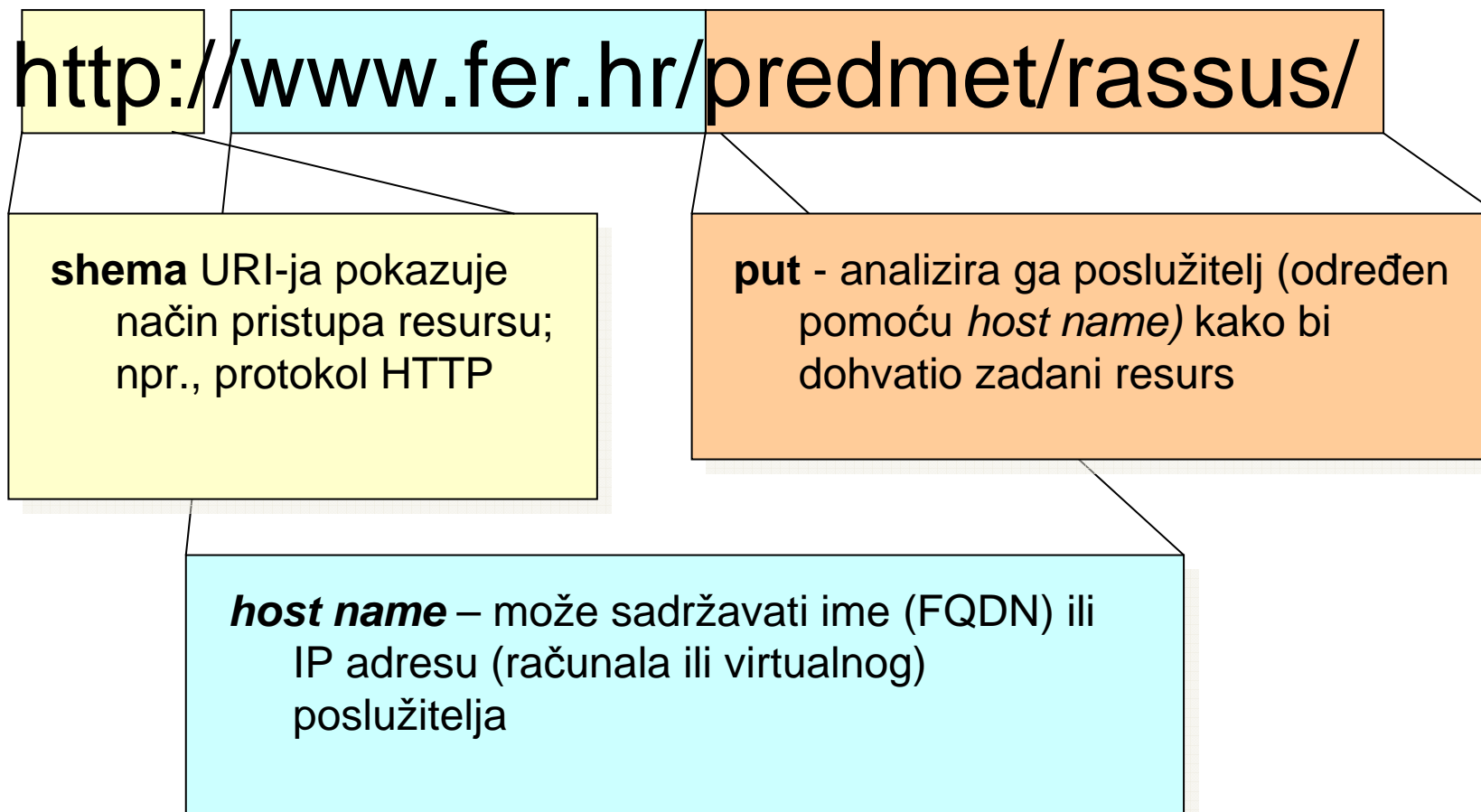
◆ URI može biti ime, lokacija ili oboje



- ◆ **poštanska adresa** je analogna URL-u
 - prof. X Y, FER, Unska 3, 10000 Zagreb, Hrvatska
 - sadrži informaciju *kamo* dostaviti poruku
 - ako se profesor Y preseli, adresa ne valja

- ◆ **jedinstveni matični broj građana (JMBG)** je analogan URN-u
 - jedinstven za osobu
 - ne sadrži uputu gdje pronaći tu osobu

Analiza uobičajenog URL-a



Opća sintaksa URI-ja (RFC 3986)



◆ Hijerarhijski niz komponenata:

- **shema** (engl. *scheme*)
- **autoritet** (engl. *authority*)
- **put** (engl. *path*)
- **upit** (engl. *query*)
- **fragment** (engl. *fragment*).

URI = **scheme** ":" hier-part ["?" **query**] ["#" **fragment**]

hier-part = "//" **authority** path-abempty
/ path-absolute
/ path-rootless
/ path-empty

- ◆ URI je složen ovako:

<shema> : <dio specifičan za shemu>

- ◆ **shema** je zaseban potprostor imena
 - npr. “**http**”, “**ftp**”, “**mailto**”, “**urn**”, “**file**”, “**news**”
 - neke sheme su dobili ime po protokolu, ali to **ne znači da je shema = protokol**
- ◆ sintaksa ostatka URI-ja posebno se definira za svaku shemu

<http://www.fer.hr/predmet/rassus/>

<http://www.w3.org/TR/webarch/#identification>

<http://www.hr/wwwhr/arts/theatre/index.hr.html>

<http://google.com/search?q=telematika>

[mailto: ime.prezime@fer.hr](mailto:ime.prezime@fer.hr)

<file:///c:/temp/>

<ftp://jdoe:jdoe@ftp.w3.org/>

<about:blank>

<urn:ietf:rfc:2396>

- ♦ većina shema *<dio specifičan za shemu>* ustrojava ovako:

// *<autoritet>* *<put>* ? *<upit>*

- ♦ **autoritet** – predstavlja (logičkog) poslužitelja
- ♦ **put** – put kroz hijerarhiju (po uzoru na datotečni sustav)
 - počinje kosom crtom i nadalje je sastavljen od segmenata odvojenih kosom crtom, npr:

/

/segment1/2/3

/mark-twain/roman/tom-sawyer

- ♦ **upit** – popis parametara u proizvoljnom redoslijedu

Shema *http*



- ◆ shema *http* definira autoritet ovako:

$\langle \text{autoritet} \rangle = \langle \text{host} \rangle : \langle \text{vrata} \rangle$

- ◆ primjer:

`http://www.fer.hr/?@=1d2w9#news_8980`

`http://www.google.com:80/search?q=telematika`

- ◆ broj **vrata** je neobavezan dio autoriteta (podrazumijeva se tcp/80)
- ◆ **put** neobavezan
- ◆ **upit** iza prvog upitnika, neobavezan
 - upit može imati više segmenata
 - segmenti upita se odvajaju znakom **&**
 - svaki segment je tipično par (ime, vrijednost) odvojen znakom **=**

- ◆ URI-ju se može dodati na kraj identifikator fragmenta entiteta odvojen znakom “#”, npr.:

`http://hr.wikipedia.org/wiki/Zagreb#Zemljopis`

- u HTML dokumentu “/Zagreb/” postoji oznaka s nazivom “Zemljopis” na čiji se vrh pozicionira prozor preglednika
- identifikator fragmenta je smislen samo ako se koristi pri akciji dohvaćanja entiteta u kojem se nalazi odgovarajući fragment

Sintaksa - posebni znakovi



- ◆ URI izravno koristi znakove iz ograničenog skupa
- ◆ ostali ASCII znakovi se predstavljaju u posebnom “*escaped*” obliku: znak “%” i dvoznamenkasti heksadekadski kôd

povećanje od +30%

- “ć” ima kôd E6, razmak 20, a “%” 25

pove%E6anje%20od%20+30%25

- kod **upita** postoji poseban način pretvaranja razmaka u “+” (kôd 2B):

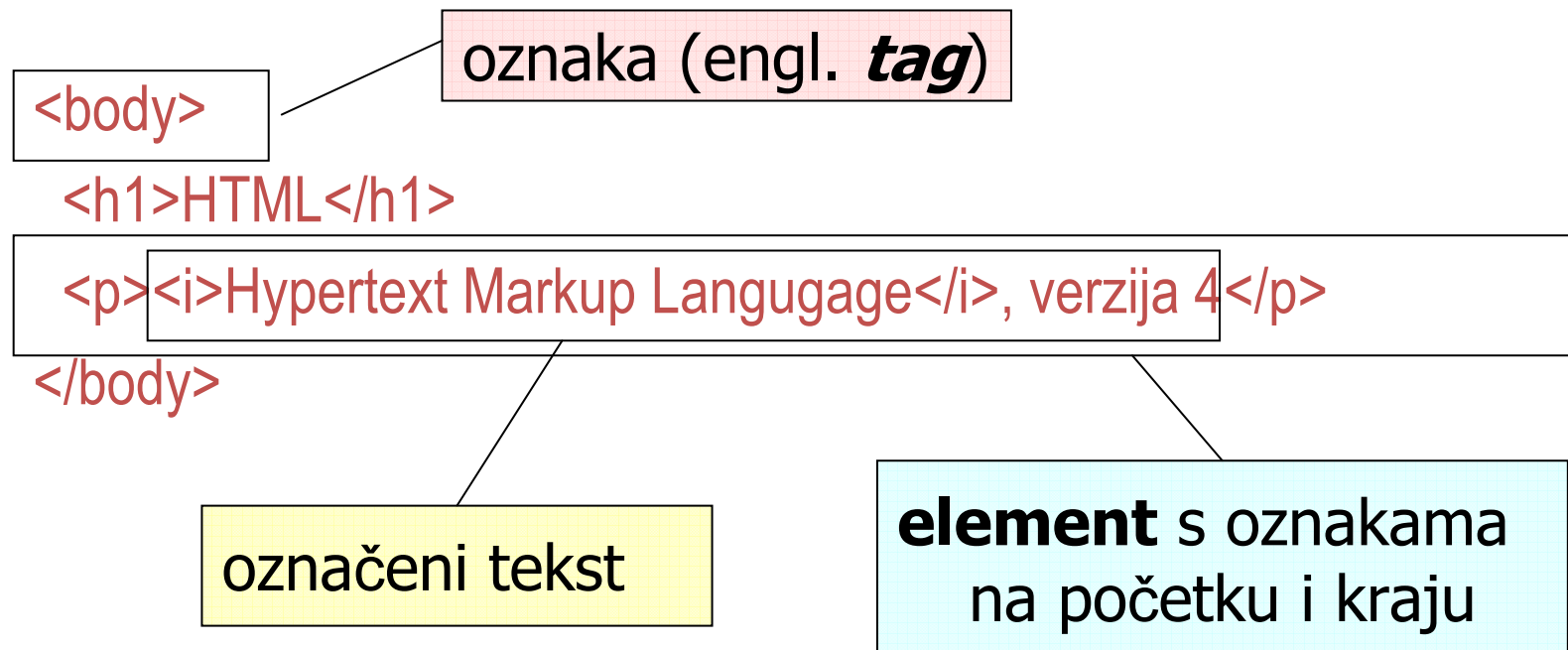
pove%E6anje+od+%2B30%25

- identifikacija izvora: URI
- zapis izvora: **HTML (XHTML)** ◀
- način povezivanja i komunikacije: HTTP

Hypertext Markup Language - HTML



- ◆ prva verzija HTML-a 1992. godine; verzija 4.01 iz 1999. (preporuka W3C-a), osnovica za *Extensible Hypertext Markup Language* XHTML
- ◆ jezik za označavanje (*markup*) – običan tekst s umetnutim oznakama koje utječu na predočavanje teksta i služe za uvođenje hiperveza



<html>

DOKUMENT

<head>

<title>TU: HTML: ustroj dokumenta</title>

<meta name="author" content="Ivo Ivic">

</head>

ZAGLAVLJE

<body>

<h1>Ustroj dokumenta u HTML-u</h1>

<p>HTML dokument sa sastoji od zaglavlja i

tijela.</p>

</body>

TIJELO

</html>

Izgled u pregledniku



<html>

<head>

<title>TU: HTML: ustroj dokumenta</title>

<meta name="author" content="Ivo Ivic">

</head>

<body>

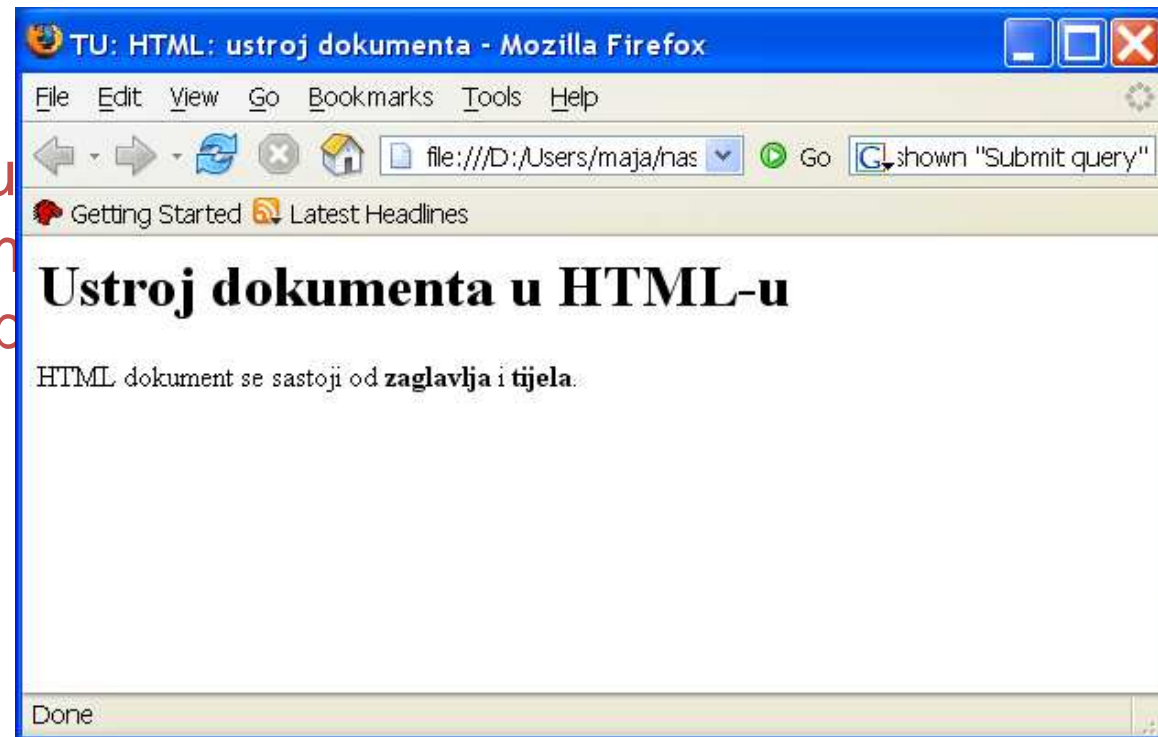
<h1>Ustroj doku

<p>HTML doku

tijela.</p>

</body>

</html>



- ◆ organizacija dokumenta
 - odjeljak (paragraf) **<p>**
 - naslov poglavlja (*heading*) **<h1>** **<h2>** ...
- ◆ izgled slova
 - podebljanje (*bold*): ****
 - kurziv (*italic*): **<i>**
- ◆ hiperveza (poveznica): **tekst**
- ◆ umetnuta slika: ****
- ◆ tablice: **<table>** **<tr>** **<td>** ...


```
<form action="http://google.com/search" method=GET>
```

Pretraživanje izraza: `<input type="text" name="q"><p>`

Broj rezultata na stranici:

```
<select name="num">
```

```
<option value="10" selected="true">10
```

```
<option value="20">20
```

```
<option value="50">50
```

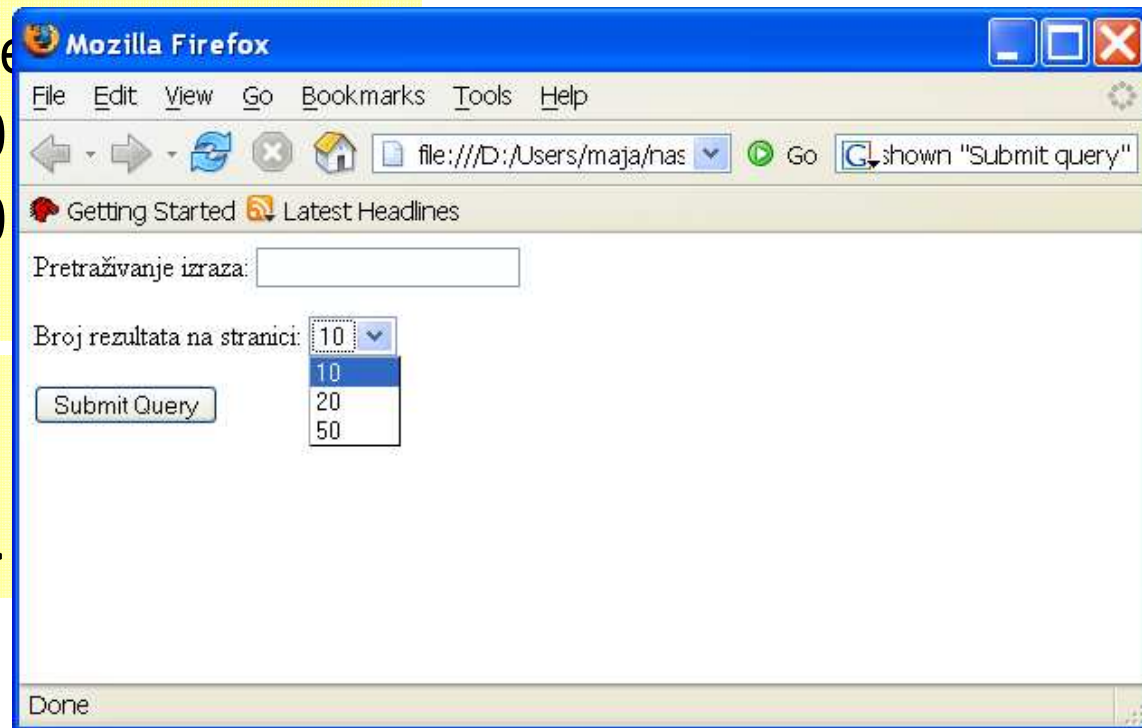
```
</select><p>
```

```
<input type="submit"
```

```
name="search"
```

```
value="Pretraži Web!">
```

```
</form>
```



- ◆ često primjenjivani formati dokumenata (na webu i drugdje)
 - umetnute slike: GIF, JPEG, PNG
 - dokumenti: PDF, Postscript
 - multimedijski dodaci: MPEG, QuickTime, WM
 - ...
- ◆ formati zasnovani na XML-u (*Extensible Markup Language*)
 - XHTML - HTML zapisan pomoću XML-a
 - vektorska grafika: SVG (*Scalable Vector Graphics*)
 - multimedijske prezentacije: SMIL (*Synchronized Multimedia Integration Language*)
 - unos elektroničkim perom: InkML (Ink Markup Language)
 -

- ◆ tehnički problemi sintakse
 - tolerancija pogrešaka, loše performanse, *namespace*-i
- ◆ pomiješan sadržaj sa izgledom
 - teško održavanje (izmjena sadržaja ili izgleda nezavisno od drugog sastojka)
 - pruža samo jedan “pogled” na podatke
- ◆ orijentiran na prikaz
 - nedostatak semantike
 - problematično pretraživanje
- ◆ nedosljedna podrška preglednika
 - odstupanje od standarda

Rješenje:
XHTML

- ◆ *Extensible Hypertext Markup Language* (XHTML™)
 - <http://www.w3.org/MarkUp/>
 - prva verzija XHTML 1.0, 1/2000, revizija 8/2002
 - u izradi verzija XHTML 2.0
- ◆ XHTML zapisuje HTML pomoću XML-a
 - novi, univerzalni standard formatiranja na Webu
 - ima strožu sintaksu → brže se parsira → brže se prikazuje u pregledniku
 - proširiv je – korisnik može dodati svoje oznake (“tagove”)
- ◆ moguć problem: nedostatak podrške u preglednicima
<http://www.w3.org/People/mimasa/test/xhtml/media-types/results>
- ◆ novi standard za obrasce: **XForms**

Neke razlike u odnosu na HTML



- ◆ najvažnije: zapis u XML-u
- ◆ osjetljivost na velika i mala slova
 - sva imena *tagova* moraju biti zapisana malim slovima
- ◆ obavezni zatvarajući *tagovi*
- ◆ svi atributi moraju imati vrijednost
- ◆ vrijednost atributa mora biti u navodnicima
- ◆ nema križanja elemenata (**<i>!!!</i>**)

Sustav upravljanja sadržajem

(CMS – *Content Management System*)

- ◆ aplikacija weba za pripremu sadržaja, automatsko generiranje navigacijskih elemenata, omogućavanje indeksiranje i pretraživanja, uspostavljenje korisničkih prava i praćenje uporabe, definiranje sigurnosnih postavki i drugo
- ◆ primjeri:
 - Joomla (www.joomla.org) – Lab 1
 - Plone (plone.org) – Lab 1
 - Quilt CMS (FER)

- identifikacija izvora: URI
- zapis izvora: HTML (XHTML)
- način povezivanja i komunikacije: **HTTP** ◀

Protokol *Hypertext Transfer Protocol (HTTP)*

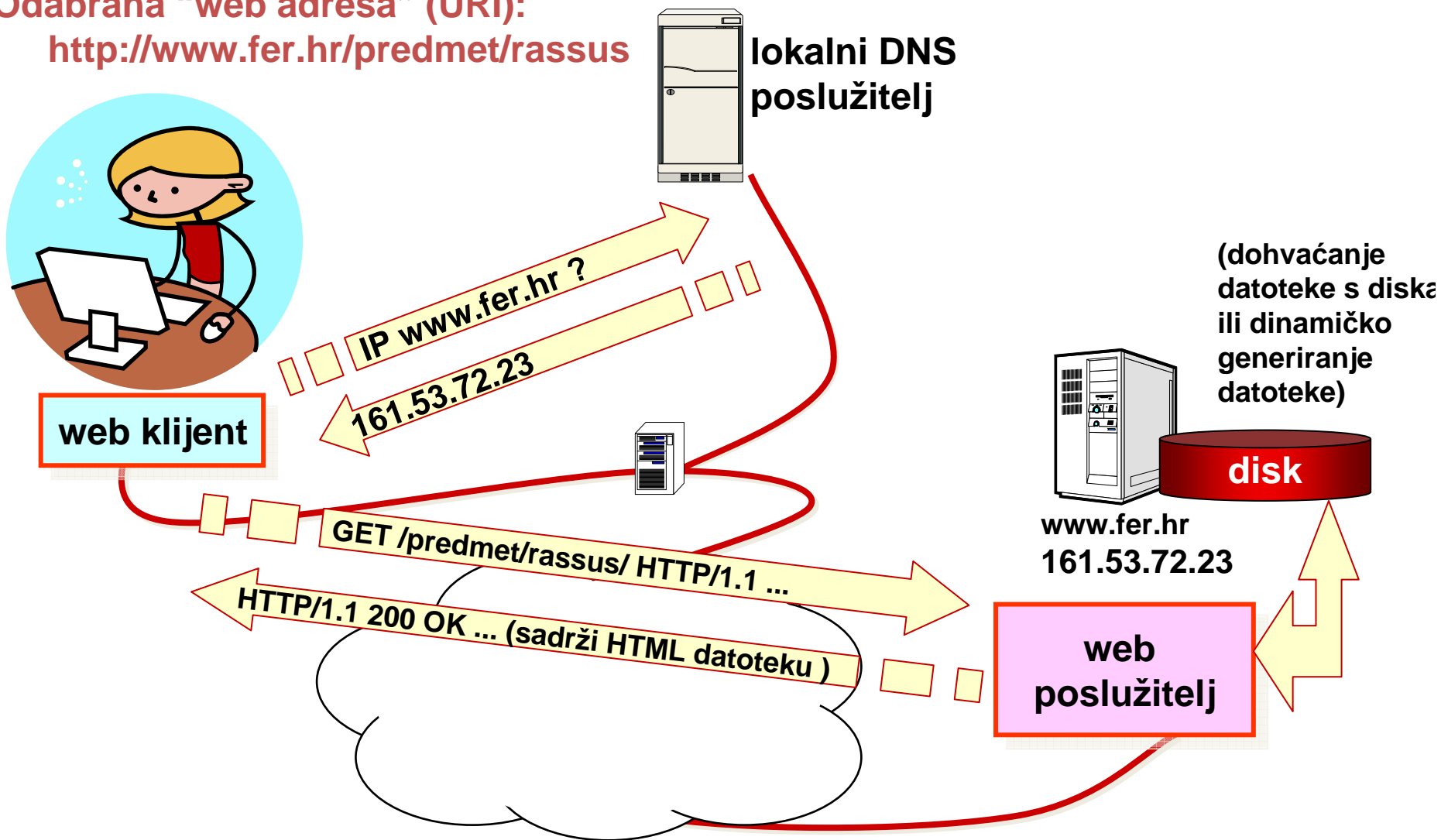


- ◆ internetski protokol aplikacijskog sloja
- ◆ definira format i način razmjene poruka
 - tekstualan zapis, sličan formatu e-mail poruke i MIME standarda
- ◆ vrste poruka:
 - **zahtjev** (“metoda” - naziv potječe od terminologije iz područja objektno-orijentiranog programiranja)
definira operaciju (metodu), resurs, protokol
 - **odgovor** (ishod zahtjeva i rezultat)
ishod zahtjeva (uspjeh, neuspjeh, greška,...) opisan statusnim kôdom
za neke vrste zahtjeva, kao rezultat uspješnog ishoda, u tijelu odgovora dostavlja se sadržaj zatraženog resursa

Komunikacija HTTP klijenta i poslužitelja



Odabrana “web adresa” (URI):
http://www.fer.hr/predmet/rassus



Komunikacija HTTP klijenta i poslužitelja (opis)



1. proces www poslužitelja (uvijek) osluškuje TCP zahtjeve na dobro-poznatim vratima 80 (ako nije drugačije konfiguriran!)
2. koristeći klijentski program (preglednik), korisnik upisuje adresu traženog izvora (**URI**)
3. preglednik saznaje IP adresu poslužitelja putem upita na DNS
4. preglednik pokreće TCP vezu sa slobodno odabranih vrata na lokalnom računalu na IP adresu poslužitelja i TCP vrata 80 (port je “dobro-poznat”)
5. nakon uspostave TCP veze, preglednik zahtijeva da mu poslužitelj pošalje dokument (**HTTP-zahtjev**)
6. poslužitelj šalje dokument(e) (**HTTP-odgovor**)
7. nakon uspješnog transfera, TCP veza se zatvara
8. preglednik prikazuje dokument (**HTML**) korisniku

Specifikacije protokola HTTP (1/2)



- ◆ prva verzija "0.9" - ograničene mogućnosti
 - podržava prijenos samo hipertekstualnih dokumenata
- ◆ **HTTP 1.0** - prvi (*Informational*) RFC (RFC 1945), svibanj 1996.
 - podržava prijenos različitih tipova podataka
 - posuđuje koncepte iz MIME standarda
 - zadržava kompatibilnost unazad
 - sredinom 90-tih - HTTP promet dominira - HTTP/1.0 neučinkovit
 - svako sjedište mora biti na drugom poslužitelju
 - preko jedne konekcije ostvaruje se jedan HTTP zahtjev
 - nema podrške za upravljanje performansama - cache, proxy, djelomični dohvat
 - WWW prozvan "World Wide Wait"

◆ HTTP verzija 1.1 - RFC 2616, lipanj 1999.

- zadržava kompatibilnost unazad prema HTTP 1.1
- RFC 2617 - autentifikacija i sigurnost

◆ poboljšanja:

- jedno fizičko računalo - više Web poslužitelja - "virtualni host"
- trajne konekcije - više zahtjeva preko jedne TCP konekcije
- djelomični dohvat sadržaja
- bolja podrška za priručna spremišta (engl. *cache*) i zastupnike(engl. *proxy*)
- pregovaranje o sadržaju datoteke (engl. *content negotiation*)
- bolji sigurnosni mehanizmi - autentifikacija

- ◆ zahtjev i odgovor moraju biti ispravno formatirani

- ◆ HTTP definira opći format poruke
 - tekstualan zapis (kao SMTP)
 - naslanja se na format e-mail poruke (RFC 822) i MIME standarda
 - dijele neka načela, ali ne sasvim i ne potpuno
 - npr. ne koriste se sva MIME zaglavlja
 - npr. tijelo ne mora biti 7-bitni ASCII

zahtjev

GET /predmet/rassus HTTP/1.1

Host: www.fer.hr

...

Accept-Language: hr, en

Accept-Encoding: gzip,deflate

...

odgovor

HTTP/1.1 200 OK

Date: Mon, 07 Apr 2008 17:31:09 GMT

Server: Apache/2.2.8 (FreeBSD) ..

Last-Modified: Mon, 30 Jan 2006
16:12:36 GMT

...

Keep-Alive: timeout=3, max=61

Connection: Keep-Alive

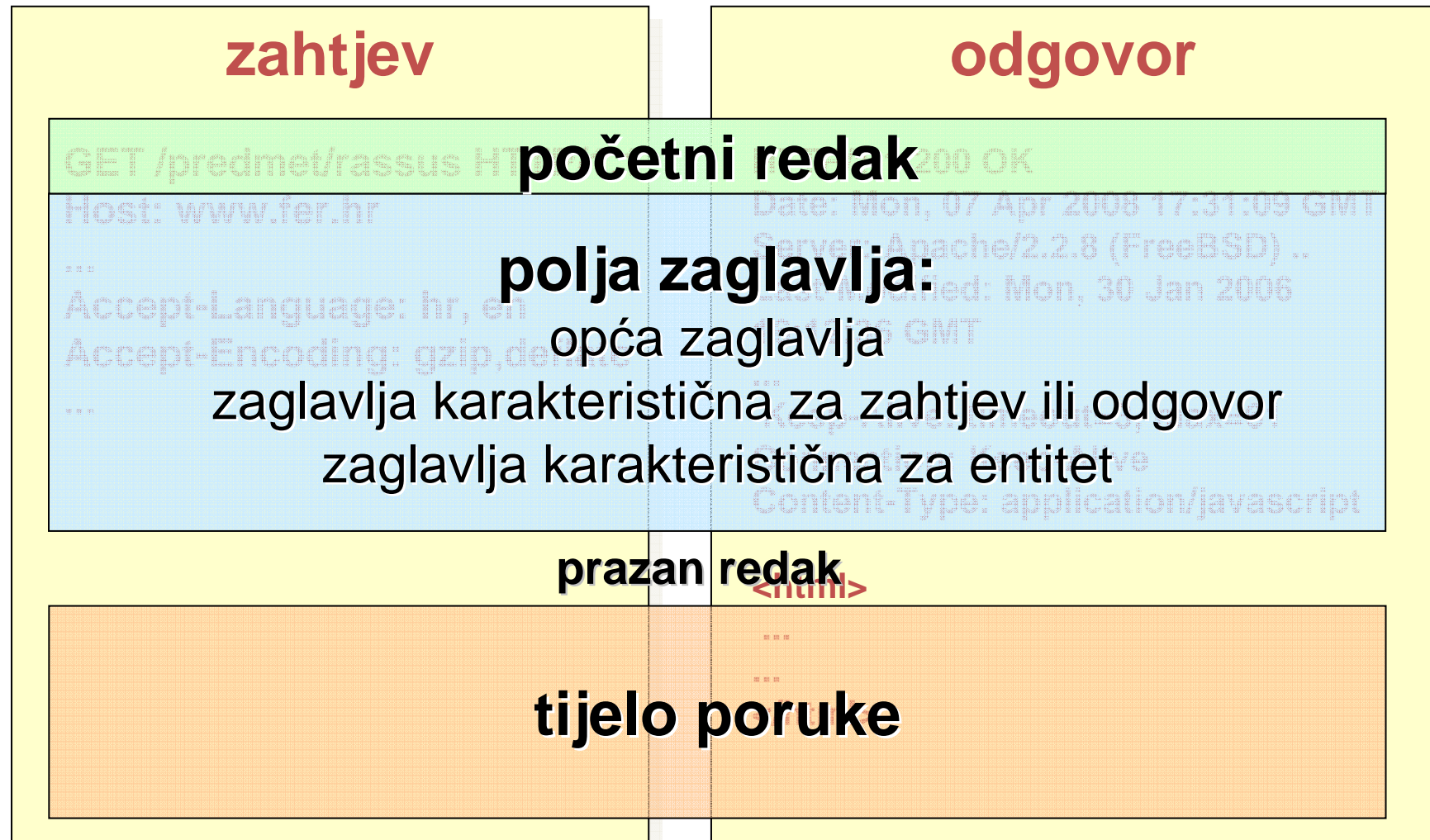
Content-Type: application/javascript

<html>

...

...

</html>



◆ početni redak sadrži (*request line*):

- resurs nad kojim je podnesen zahtjev
- metodu (operacija) se traži nad tim resursom
- verziju protokola koristi

<*metoda*> <*URI*> <*verzija*>

◆ primjeri:

- GET / HTTP/1.0
- POST /shop/order HTTP/1.1
- HEAD /search?q=raspored HTTP/1.0

- ◆ metoda zahtjeva određuje što se traži od resursa
- ◆ HTTP/1.1 definira 8 metoda i omogućuje dodavanje novih metoda (*extensions*):
 - OPTIONS
 - GET
 - HEAD
 - POST
 - PUT
 - DELETE
 - TRACE
 - CONNECT
- ◆ važna svojstva: sigurnost i idempotentnost

GET – metoda za dohvaćanje

- znači “Pribavi reprezentaciju tog resursa”
- aktivira se kod upisivanja adrese u preglednik ili klika na link

- ◆ ako poslužitelj ima zahtijevani resurs, vraća ga u tijelu odgovora, inače vraća pogrešku
- ◆ ponašanje metode GET se mijenja ako se koristi uvjetni GET (zaglavlje *If-Modified-Since:* ili *If-Match:*)
- ◆ moguće koristiti djelomični GET (partial GET) - dohvaća se samo dio datoteke (u zaglavlju zahtjeva definiran raspon *Range*)

Primjer primjene GET metode



GET / HTTP/1.0

prazan redak, bez tijela

GET /a/b?c HTTP/1.1
Host: www.tel.fer.hr

GET /obicni.txt HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-excel, application/vnd.ms-powerpoint,
application/msword, */*
Accept-Language: hr
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
Host: www.tel.fer.hr
Connection: Keep-Alive

HEAD – metoda za dohvaćanje podataka o resursu

- razlika u odnosu na GET: poslužitelj **ne** vraća sadržaj resursa u tijelu odgovora
- najčešće se koristi kako bi se provjerilo postoji li entitet na poslužitelju
- ◆ druge uporabe:
 - provjera veličine datoteke prije dohvaćanja
 - pribavljanje metapodataka o entitetu
- ◆ na poslužitelju se HEAD zahtjev obrađuje jednako kao i GET

POST – metoda za “aktiviranje” resursa

- znači “Pomoću adresiranog resursa obradi podatke koje šaljem”
- obično se aktivira se pritiskom na gumb u obrascu

- ◆ može se koristiti kod ispunjavanja web-obrasca
 - podaci koje je upisao korisnik, prenose se metodom POST na poslužitelj i tamo se obrađuju
 - primjeri:
 - zahtjev za provedbom narudžbe
 - dodavanje vlastitog komentara tekstu na Webu
 - prijavljivanje na termin laboratorijskih vježbi
 - u ovom primjeru, koristio bi se gumb “*Submit*” na obrascu

Primjer primjene metode POST



POST /search HTTP/1.1

Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg...

Accept-Language: hr

Content-Type: application/x-www-form-urlencoded

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)

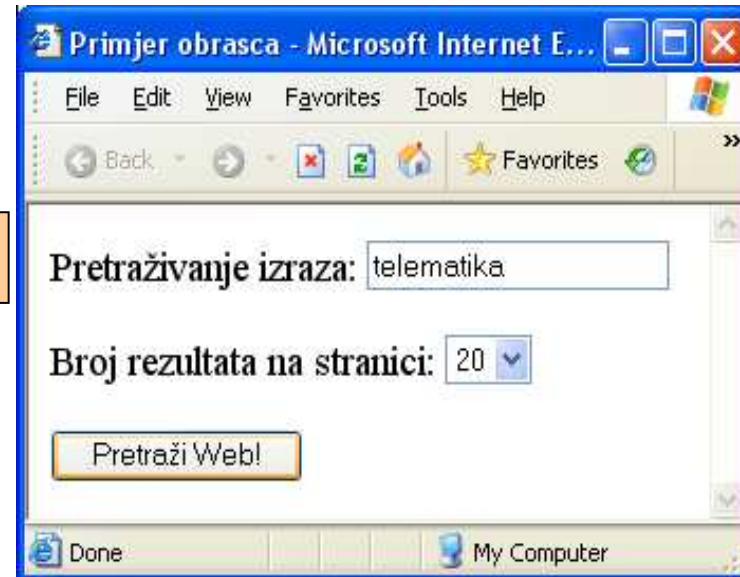
Host: localhost:4744

Content-Length: 44

Connection: Keep-Alive

Cache-Control: no-cache

q=telematika&num=20&search=Pretra%9Ei+Web%21



OPTIONS

- informiranje o mogućnostima resursa i poslužitelja (URI = *)

PUT

- postavljanje entiteta uključenog u tijelu zahtjeva na zadani URI
 - POST u URI navodi resurs koji mora obraditi podatke koji se šalju u tijelu zahtjeva
 - PUT navodi naziv resursa u kojeg treba pohraniti podatke koji se šalju u tijelu
- najčešće se ne koristi - sigurnosni rizik

DELETE

- brisanje odabranog resursa - ne koristi se

TRACE

- klijent dobiva kopiju zahtjeva kojeg je uputio poslužitelju, služi za dijagnostiku

CONNECT (za buduću uporabu)

◆ općenita zaglavlja

- odnose se na poruku, a ne na njen sadržaj
- mogu se pojaviti i u zahtjevu i u odgovoru

◆ zaglavlja karakteristična za zahtjev

- poslužitelju daju više informacija o prirodi zahtjeva
- omogućuju klijentu da kontrolira kako će se zahtjev ostvariti
 - npr. uvjetni zahtjev - samo ako su ispunjeni određeni uvjeti
- klijent obavještava poslužitelja koje vrste podataka može obraditi

◆ zaglavlja karakteristična za entitet

- opisuju entitet koji se nalazi u tijelu poruke, ako ga ima

Zaglavlja zahtjeva

METHOD:

- GET
- POST
- HEAD
- PUT
- OPTIONS
- TRACE
- CONNECT
- DELETE

Method Request-URI HTTP-Version

parametar1: *vrijednost*

parametar2: *vrijednost*

parametar3: *vrijednost*

....

<prazni redak>

HTTP/1.0
HTTP/1.1

general_header

Cache-Control:
Connection:
Date:
Pragma:
Trailer:
Transfer-Encoding:
Upgrade:
Warning:

request_header

Accept:	If-Modified-Since:
Accept-Charset:	If-None-Match:
Accept-Encoding:	If-Range:
Accept-Language:	If-Unmodified-Since:
Authorization:	Max-Forwards:
Expect:	Proxy-Authorization:
From:	Range:
Host:	Referer:
If-Match:	TE:
	User-Agent:

entity_header

Allow:
Content-Encoding:
Content-Language:
Content-Length:
Content-Location:
Content-MD5:
Content-Range:
Content-Type:
Expires:
Last-Modified:
extension-header

◆ početni redak sadrži:

■ verziju protokola

■ statusni kôd

■ opisnu frazu

HTTP/1.1 303 See Other

HTTP/1.1 200 OK

HTTP/1.1 404 Not Found

◆ u **tijelu** odgovora se obično prenosi reprezentacija resursa (“entitet”) koju preglednik treba prikazati korisniku

◆ neka polja zaglavlja:

■ **Content-Type**: format entiteta

■ **Content-Length**: duljina entiteta u tijelu u oktetima

- ◆ opća zaglavlja
 - odnose se na poruku, a ne na njen sadržaj
 - mogu se pojaviti i u zahtjevu i u odgovoru

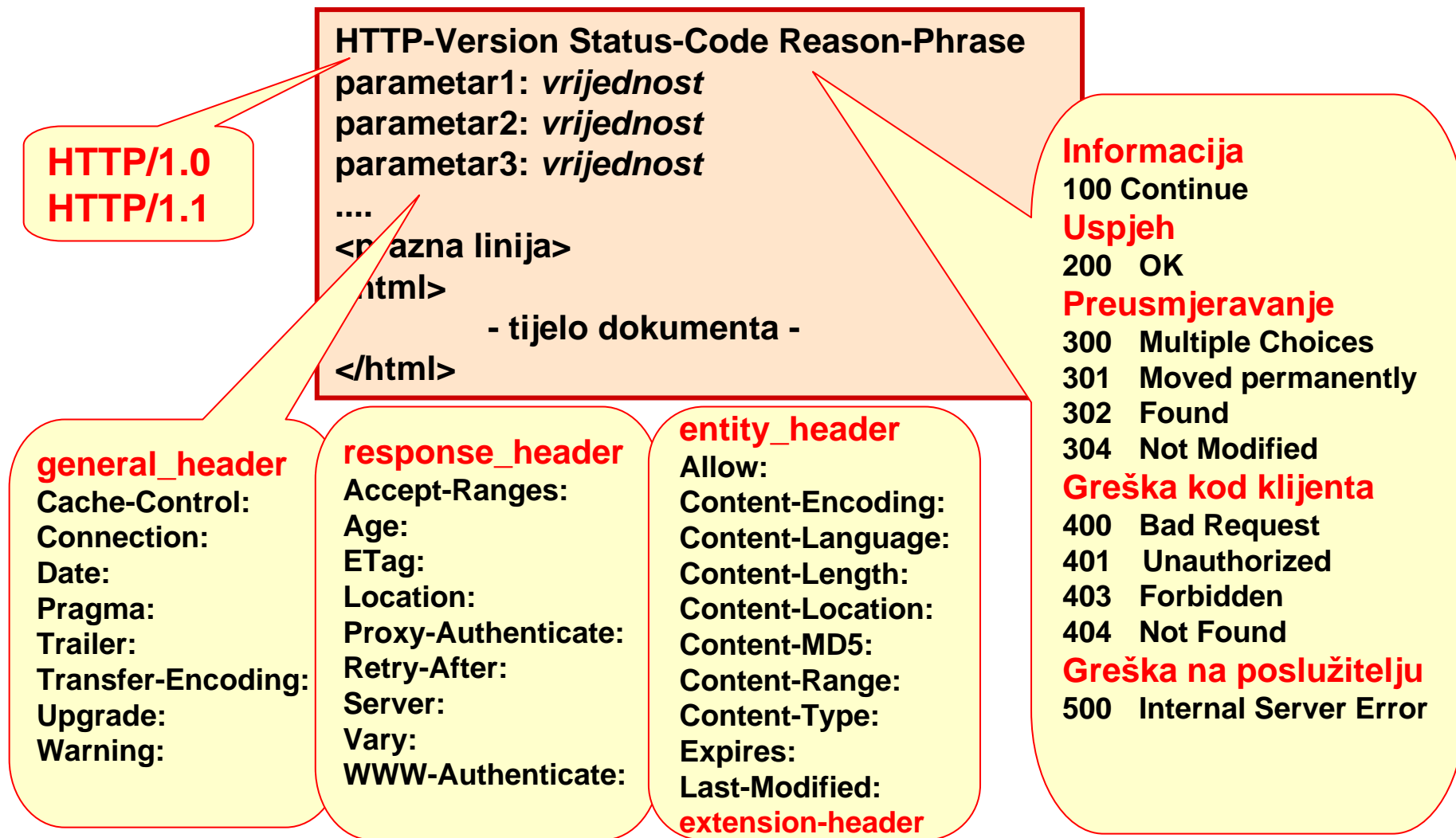
- ◆ zaglavlja karakteristična za odgovor
 - klijentu daju više informacija o odgovoru
 - dio informacija može se prenijeti i u tijelu poruke (npr. opis pogreške)

- ◆ zaglavlja karakteristična za entitet
 - opisuju entitet koji se nalazi u tijelu poruke, ako ga ima
 - češće se koriste kod odgovora nego kod zahtjeva

- ◆ sastoji se od tri dekadске znamenke

- ◆ pet kategorija:
 - 1xx – **Informativne** - ne naznačuju ni uspjeh, ni neuspjeh
 - 2xx – **Uspjeh** - poslužitelj je primio, razumio i ispunio zahtjev
 - 3xx – **Preusmjeravanje** - potrebno poduzeti dodatne akcije
 - 4xx – Greška na **klijentu** - zahtjev je neispravan
 - 5xx - Greška na **poslužitelju** - zahtjev je ispravan, ali poslužitelj ga ne može ispuniti

Zaglavlja odgovora



Uspješan odgovor (2xx)



- ◆ poslužitelj je uspješno primio, razumio i ispunio zahtjev
- ◆ najčešći kôd **200** (*OK*)
 - za zahtjev **GET** znači da je dostavljeni entitet u tijelu odgovora sadržaj resursa
 - za **POST** znači da je resurs primio podatke i dostavljeni entitet opisuje ishod akcije
- ◆ najčešći odgovor na Webu

Drugi kôdovi uspješnog odgovora

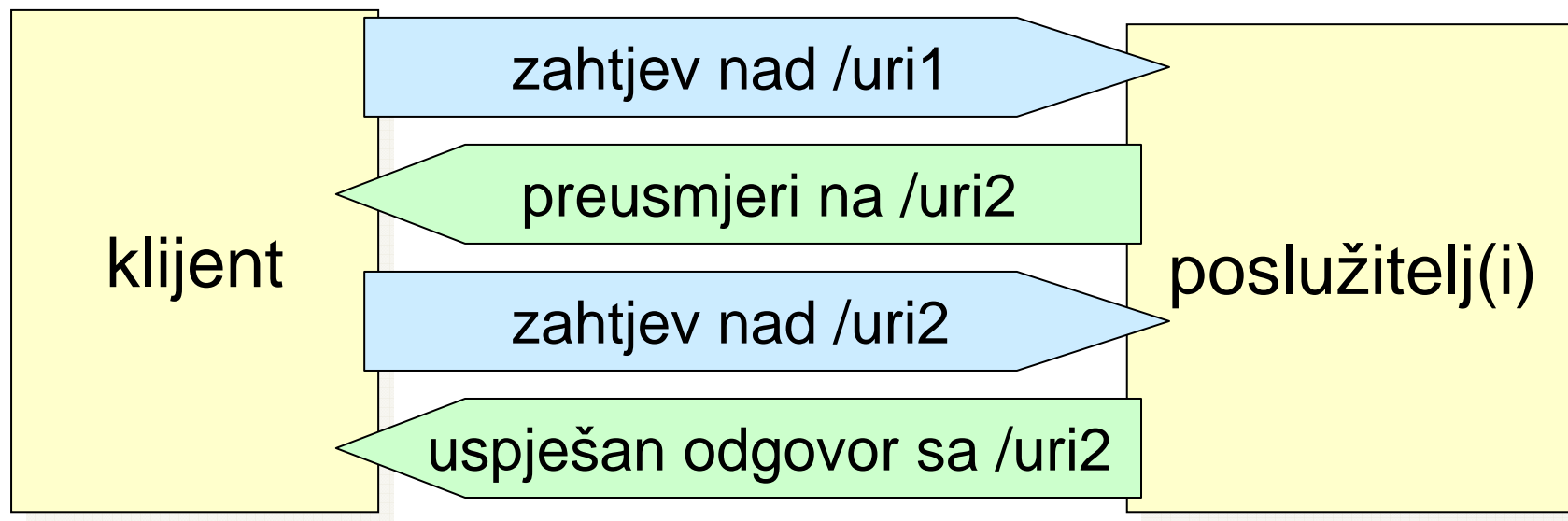


- ◆ 201 (Created) - rezultat metode PUT
- ◆ 204 (*No Content*)
 - poslužitelj je ispunio zahtjev, ali nema potrebe da vraća ikakav entitet u tijelu odgovora (npr. anketa)
 - korisnički agent ne mijenja prikaz prošlog entiteta
- ◆ 205 (*Reset Content*)
 - korisnički agent treba sadržaj postojećeg entiteta postaviti na početne vrijednosti
 - namijenjeno za višestruki unos podataka preko obrasca
- ◆ 206 (*Partial Content*)
 - vraćena je djelomična manifestacija resursa
 - zahtjev je sadržavao polje zaglavlja Range:

Preusmjeravanje (3xx)



- ◆ klijent treba poduzeti dodatne korake kako bi ispunio izvorni zahtjev
- ◆ novi URI se nalazi u polju zaglavlja **Location**:
- ◆ sigurne metode se mogu izvršiti bez sudjelovanja korisnika
 - sigurne: GET, HEAD, OPTIONS, TRACE



Kôdovi premještanja (301 i 307)

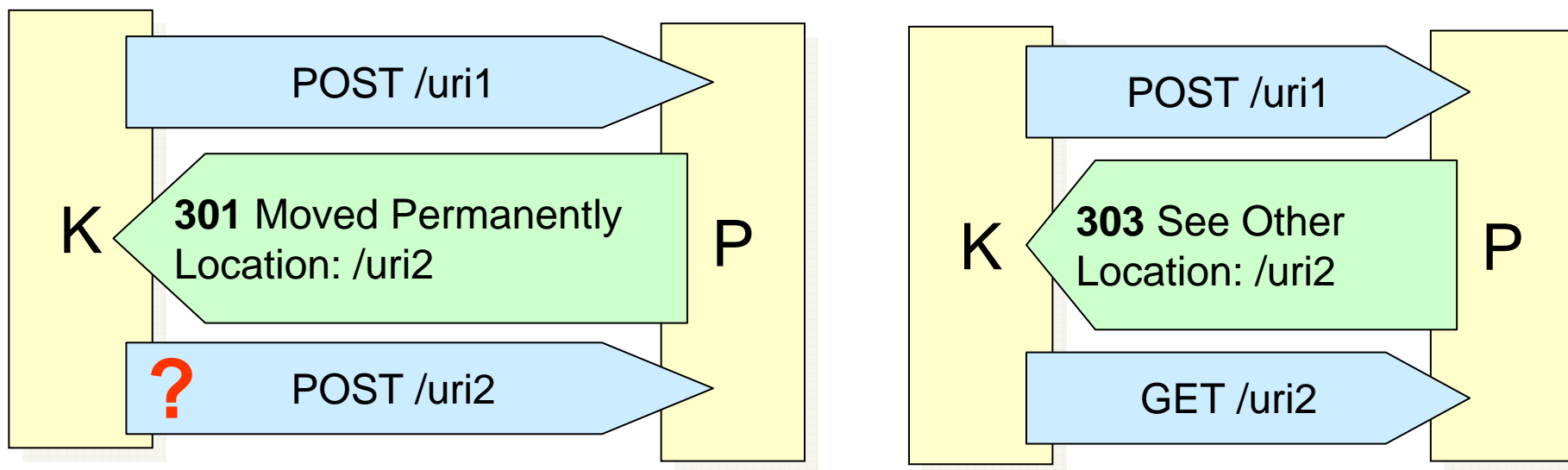


- ◆ ponekad se resursu dodjeli drugi URI, pa se poslužitelj može konfigurirati da preusmjerava zahtjeve na novu adresu
- ◆ 301 (*Moved Permanently*)
 - resurs je trajno premješten i svi bi budući zahtjevi trebali biti usmjereni na novi URI
- ◆ 302 (*Found*)
 - resurs privremeno koristi drugi URI, klijent i dalje može koristiti stari URI
- ◆ 307 (*Temporary Redirect*)
 - resurs je samo privremeno premješten, pa se treba sačuvati stara adresa
- ◆ klijent ponavlja identičan zahtjev nad novim URI-jem
- ◆ ako metoda nije sigurna, traži se potvrda korisnika
- ◆ u tijelu odgovora se može nalaziti poruka za “ručno” preusmjeravanje

303 (See Other)



- ◆ zahtjev je ispunjen, ali sada treba prikazati sadržaj nekog drugog resursa (treba izvršiti **GET** nad resursom identificiranom u polju **Location**)
- ◆ budući da je GET siguran, preusmjerenje se vrši automatski



304 (Not Modified)



- ◆ **uvjetni** (*conditional*) GET

- ◆ koristi polja zaglavlja

 - If-Modified-Since:

 - If-Unmodified-Since:

 - If-Match:

 - If-Unmatch:

- ◆ **304** (*Not Modified*)

 - traženi resurs nije promijenio sadržaj od zadnjeg zahtjeva

 - klijent može koristiti kopiju entiteta iz *cachea*

Pogreške klijenta (4xx)



- ◆ namijenjena za slučajeve kad se čini da je pogreška nastupila na klijentovoj strani
- ◆ odgovor treba sadržavati poruku namijenjenu korisniku u kojem se opisuje situacija i nude rješenja, npr:
- ◆ **400** (*Bad Request*)
 - pogreška u sintaksi zahtjeva
- ◆ **401** (*Unauthorized*)
 - zahtjevu nedostaje autorizacija korisnika
- ◆ **404** (*Not Found*)
 - resurs nije dostupan, ali se ne ulazi u detalje zašto
 - obično je pogreška pri utipkavanju URI-ja

Autorizacija (kôd 401)



- ◆ **401** (*Unauthorized*)
- ◆ zahtjevu nedostaje autorizacija korisnika
- ◆ odgovor sadrži izazov klijentu u polju **WWW-Authenticate:**
- ◆ klijent ponavlja zahtjev sa dodanim poljem **Authorization:** koje sadrži npr. šifrirano korisničko ime i zaporku
- ◆ poslužitelj izvršava zahtjev ako on sadrži potrebnu autorizaciju
- ◆ nije isto što i autorizacija preko Cookieja (ZZT Web)!

◆ 403 (Forbidden)

- poslužitelj odbija ispuniti taj zahtjev, bez obzira na autorizaciju
- primjer: klasični poslužitelj nema pristup datoteci

◆ 405 (*Method Not Allowed*)

- nije dozvoljena tražena metoda nad tim resursom

◆ 410 (*Gone*)

- resurs na tom URI-ju nije više dostupan i neće ni biti, pa je uzaludno ponavljati zahtjeve

Pogreške poslužitelja (5xx)



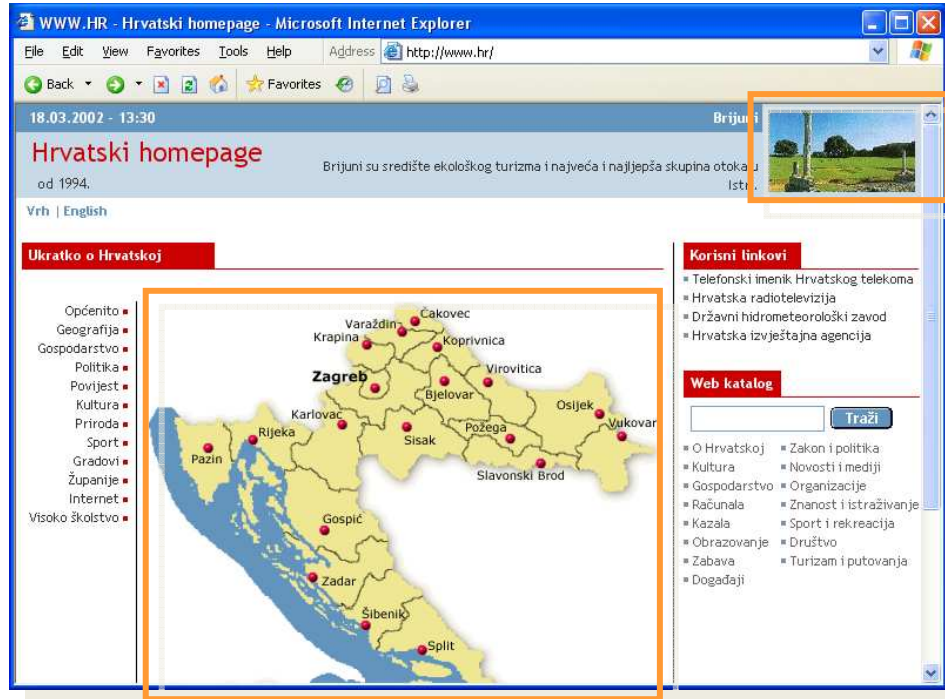
- ◆ kod ovakvih odgovora poslužitelj je “svjestan” da zahtjev nije ispunjen zbog njegove greške
- ◆ **500** (*Internal Server Error*)
 - obično programska greška u resursu
- ◆ **503** (*Service Unavailable*)
 - poslužitelj je preopterećen
 - odgovor može sadržavati **Retry-After:** polje

- ◆ HTTP-ov mehanizam izbora odgovarajuće prezentacije sadržaja resursa
- ◆ klijent predlaže prihvatljive oblike
 - polja zaglavlja: **Accept**, **Accept-Charset**, **Accept-Encoding**, **Accept-Language**, **User-Agent**
- ◆ primjer:
 - resurs podržava višejezične reprezentacije, uključujući i onu na hrvatskom
 - klijent predlaže **Accept-Language: hr**
 - poslužitelj vraća entitet na hrvatskom jeziku

- ◆ HTTP/1.1 omogućuje postojane konekcije (engl. *persistent connection*)
 - **Connection: Keep-Alive**

- ◆ dva načina korištenja konekcije:
 - *pipelining* – pošalju se svi zahtjevi i tek onda se čekaju odgovori
 - *chaining* – novi zahtjev se šalje tek nakon primitka odgovora

Primjer uporabe *pipelininga*



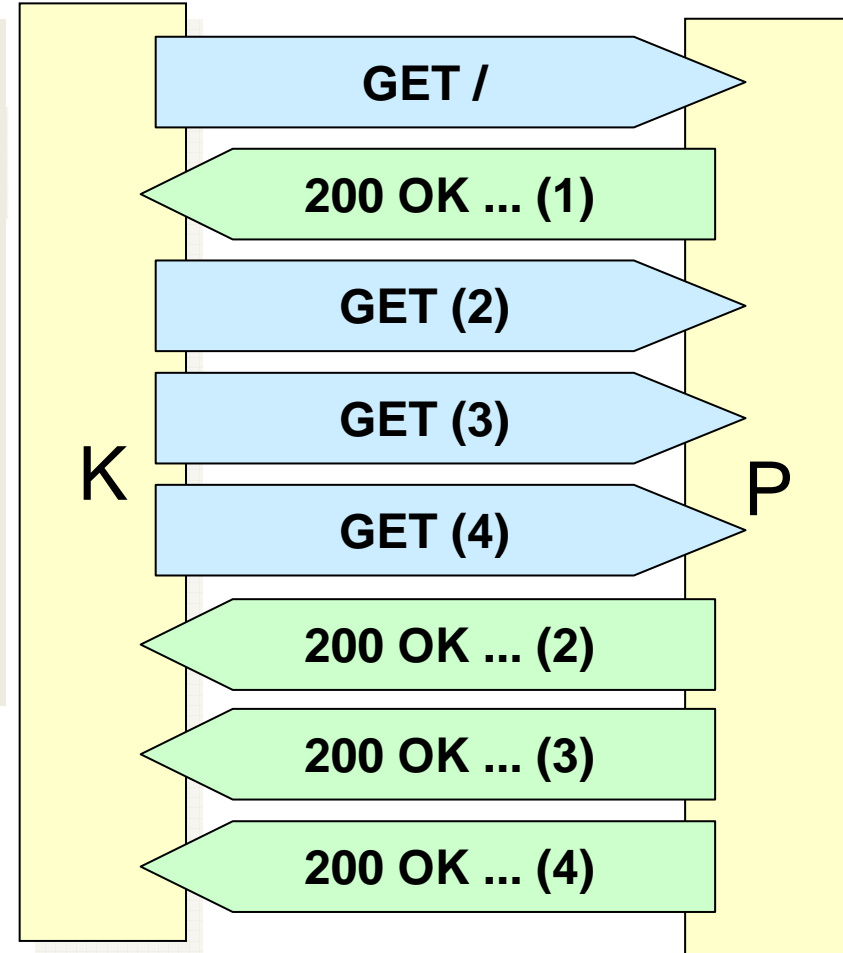
(1) <http://www.hr>

za prikaz su potrebni 2, 3 i 4

(2) <http://www.hr/style.css>

(3) <http://www.hr/hrvatska/image/hrvatska.gif>

(4) <http://www.hr/hrvatska/image/banners/brijuni1.jpg>



◆ World Wide Web Consortium (W3C)

<http://www.w3c.org/>

“The World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential.”

◆ Internet Engineering Task Force

<http://www.ietf.org>

■ RFC-Editor

<http://www.rfc-editor.org>

■ RFC Search

<http://www.rfc-editor.org/rfcsearch.html>

■ RFC Index

<http://www.rfc-editor.org/rfc-index.html>