

ՅՈՒՆԵՍԿՈ ԵՎ ՆԻԿՅՈՒՆԵՎԻՆ

ՀԱՄԱԲԱՆԱԴՐՈՒԹՅԱՆ ԵՎ ՆԻԿՅՈՒՆԵՎԻՆԻ
ԲԱՆԱԴՐՈՒԹՅԱՆ ԵՎ ՆԻԿՅՈՒՆԵՎԻՆԻ

ՄԻԱՅՈՒՆԵՎԻՆ

Otvoreno računarstvo

Komunikacija preglednika i poslužitelja

- Preglednik
- Protokol HTTP
- Poslužitelj - CGI

Mario Žagar



The background of the slide features large, light green, stylized letters 'O' and 'R' that are partially visible, creating a decorative frame. The letter 'O' is on the left and the letter 'R' is on the right, both rendered in a clean, sans-serif font. The text 'Preglednik' is centered between them.

Preglednik

Preglednik Weba



- Samostalna aplikacija za pristup uslugama
 - Domenska imena (protokol: DNS, resursi: IP adrese)
 - Web (protokol: HTTP/TCP, resursi: HTML, CSS, slike, ...)
 - FTP (protokol: FTP/TCP, resursi: datoteke)
 - RSS (protokol: HTTP/TCP, resursi: XML)
 - Streaming (protokol: RTSP, ...)
- Ugrađena podrška za interpretaciju raznih oblika zapisa
 - Statičkih: HTML, CSS, JPEG, PNG, GIF, SVG, ...
 - Dinamičkih: DHTML, JavaScript, ECMAScript, ...
 - Generiranih: XML+XSL, XSLT, ...
 - Vanjskih - podrška dodana u obliku raznih dodataka (pluginova): Flash, PDF, SVG, DjVu, Java, ...

Preglednik kao klijent



- Gdje sve možemo naći preglednik?

- Osobno računalo (PC)
- Dlanovnik (PDA)
- Mobitel
- Televizor
- Igraća konzola
- ...
- Frižider?



LG Digital Multimedia Side-By-Side Fridge Freezer

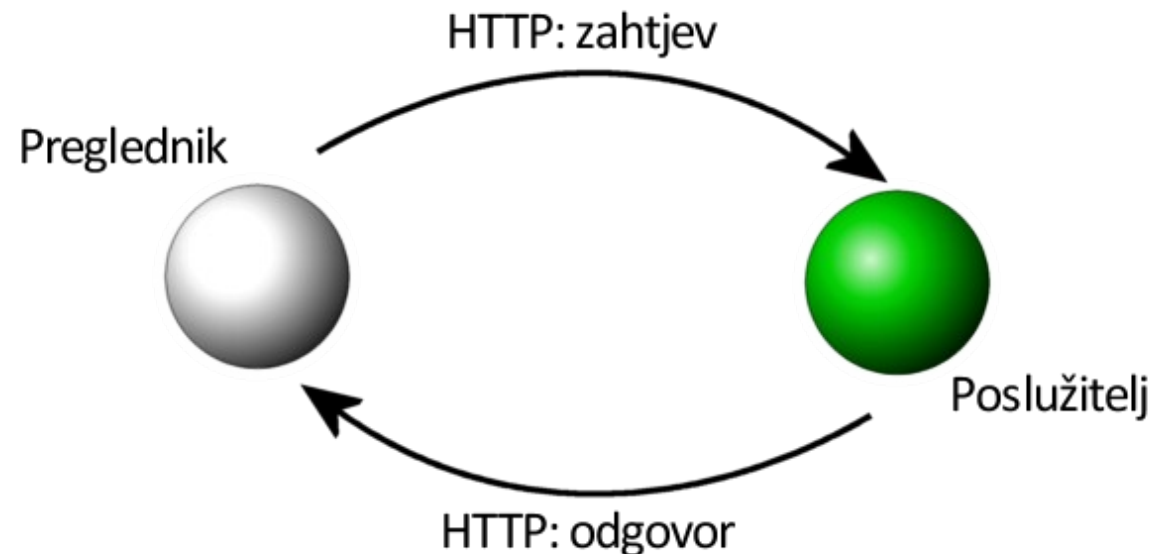
- Što mora imati uređaj koji sadrži preglednik?

- Mogućnost povezivanja s poslužiteljem

Preglednik kao klijent



- Preglednik Weba komunicira s poslužiteljem
 - Uloga *klijenta* u odnosu poslužitelj-klijent (*client-server*)
- Na koji način komunicira?
 - Protokol HTTP



Rad preglednika

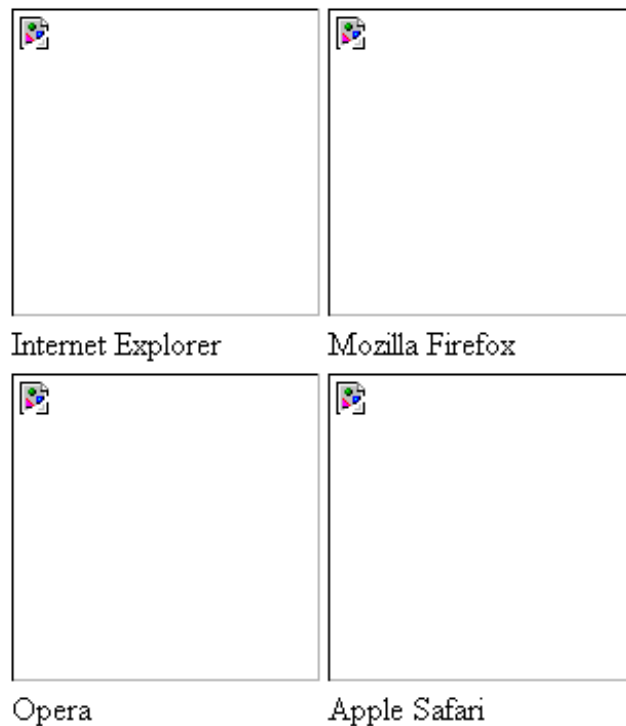
- Preglednik dohvaća objekte
 - Adresa objekta – URI
 - Protokol pristupa objektima – opisan URI-jem
- Preglednik prikazuje objekte
 - Određene oznakom MIME tipa objekta (npr. *text/html*)
 - Tumačenje sadržaja objekta radi preglednik ili dodatak pregledniku
- Objekti mogu sadržavati reference (URI) na druge objekte

Primjer – stranica Weba

- Sastoji se od nekoliko objekata
- HTML
 - URI:
`http://www.rasip.fer.hr/primjer.html`

Preglednici

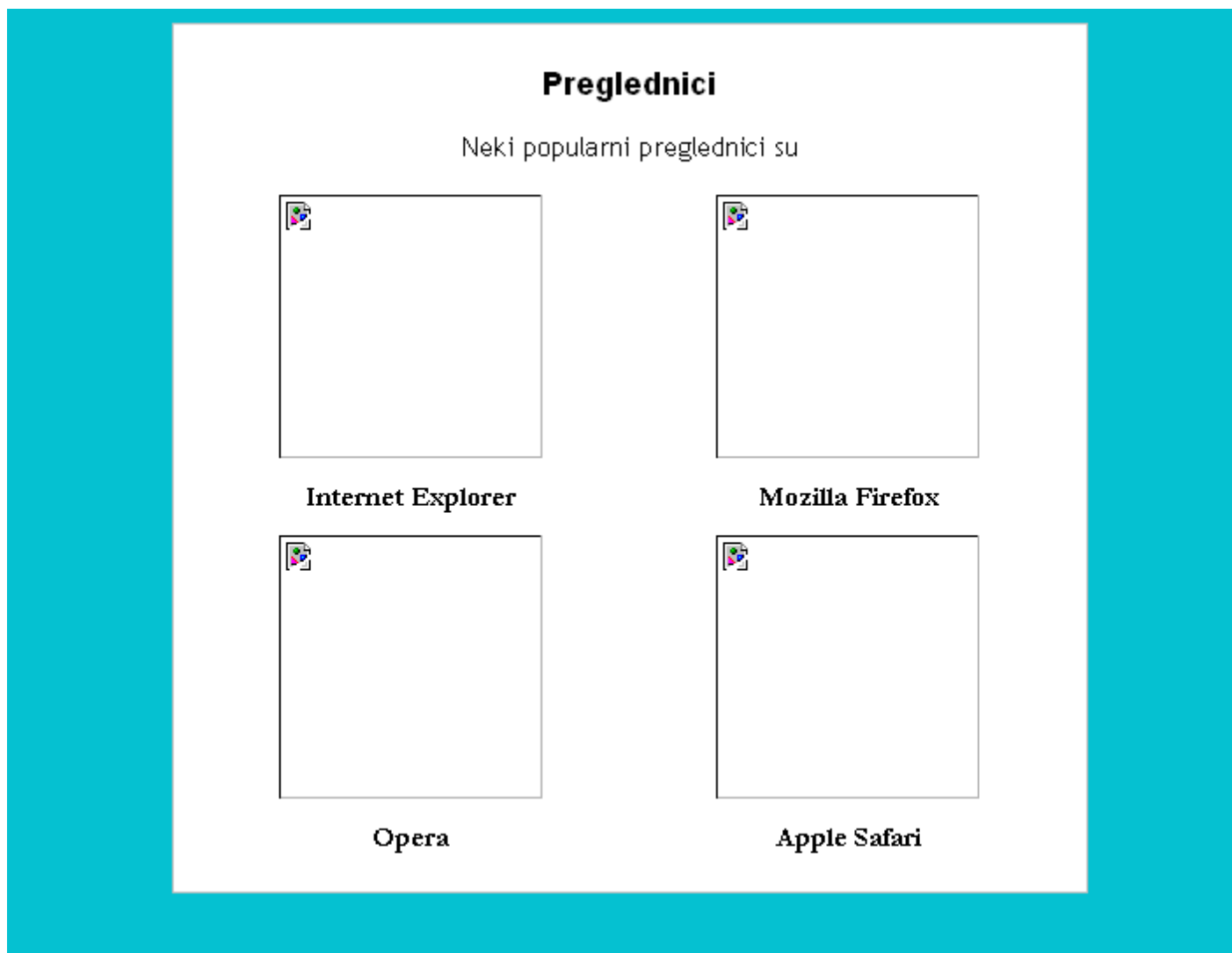
Neki popularni preglednici su



Primjer – stranica weba



- CSS – URI:<http://www.rasip.fer.hr/primjer.css>



Primjer – stranica Weba



- Slike



Dohvaćanje objekata



HTML (text/html)

<http://www.rasip.fer.hr/primjer.html>

CSS (text/css)

<http://www.rasip.fer.hr/primjer.css>

PNG (image/png)

<http://www.microsoft.com/ie7.png>

JPEG (image/jpeg)

<http://www.opera.com/opera9.jpeg>



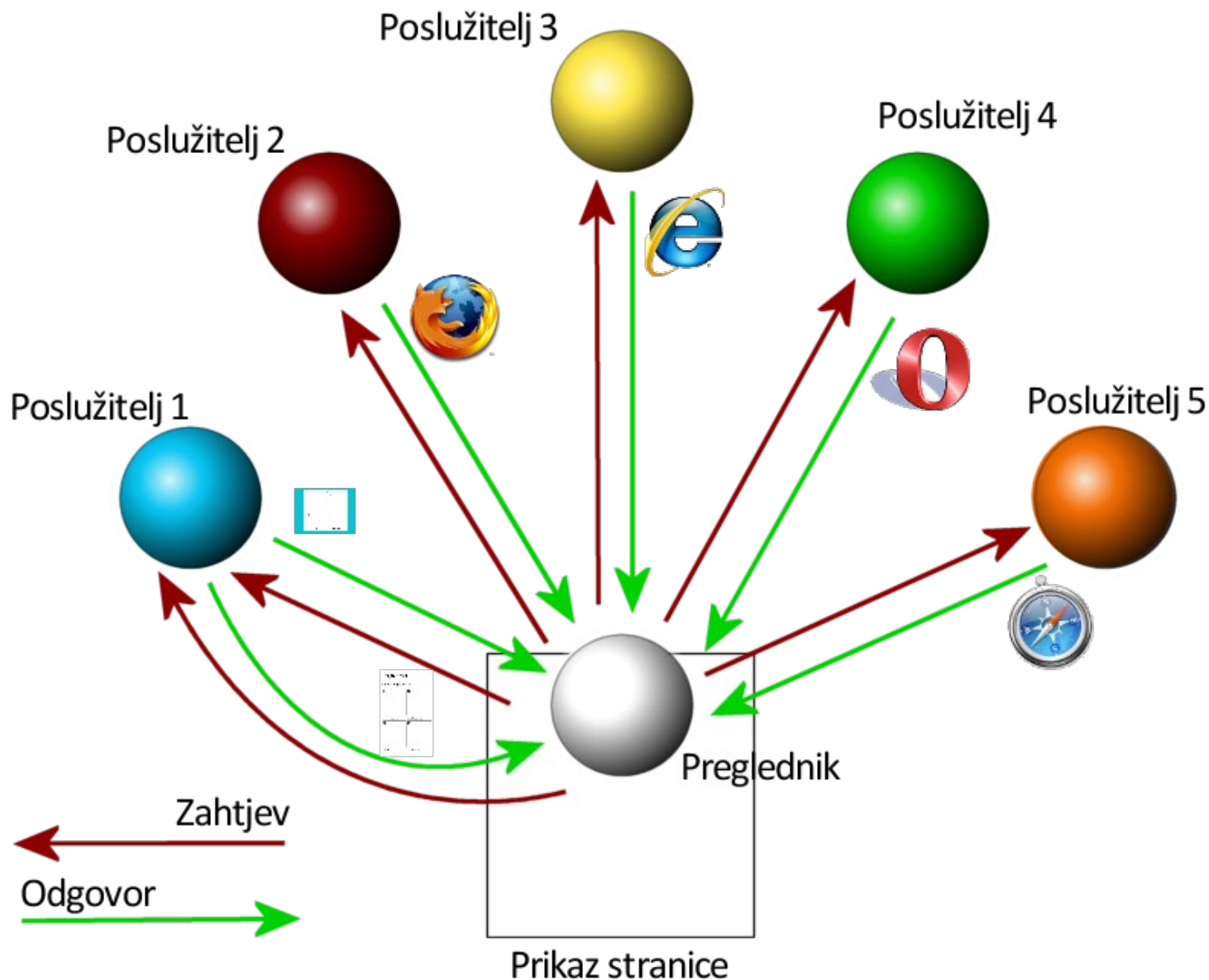
PNG (image/png)

<http://www.apple.com/safari.png>

PNG (image/png)

<http://www.mozilla.com/firefox.png>

Dohvaćanje sastavnica stranice



Mogućnost izbora

- Preglednici raznih proizvođača
 - Microsoft, Mozilla, Opera, ...
- Različite verzije istog proizvoda
 - Microsoft Internet Explorer 5, 5.5, 6, 7, 8, ...
 - Mozilla Firefox 1.0.x, 1.5.x, 2.0.x, 3, ...
 - Opera 7, 8, 9, ...
- Različita podrška za oblike zapisa (npr. PNG, CSS2)
- Različito tumačenje istih zapisa
 - Najvidljivije kod tumačenja CSS-a

Primjer: PNG

- Portable Network Graphics – PNG (čit. ping)
- Nasljednik GIF-a
 - Kompresija bez gubitaka podataka (kao i GIF)
 - Podržana 24-bitovna boja (kod GIF-a 8 bita)
 - Podržana transparentcija kroz alfa-kanal (kod GIF-a se jedna boja može odrediti kao transparentna)
 - Otvorena norma (autorska prava na LZW kompresiju korištenu u GIF-u drži Unisys)

Primjer: PNG

- *Podrжан* u svim zastupljenijim preglednicima
- Problemi: npr. transparentcija nije u potpunosti implementirana u MS IE sve do verzije 7

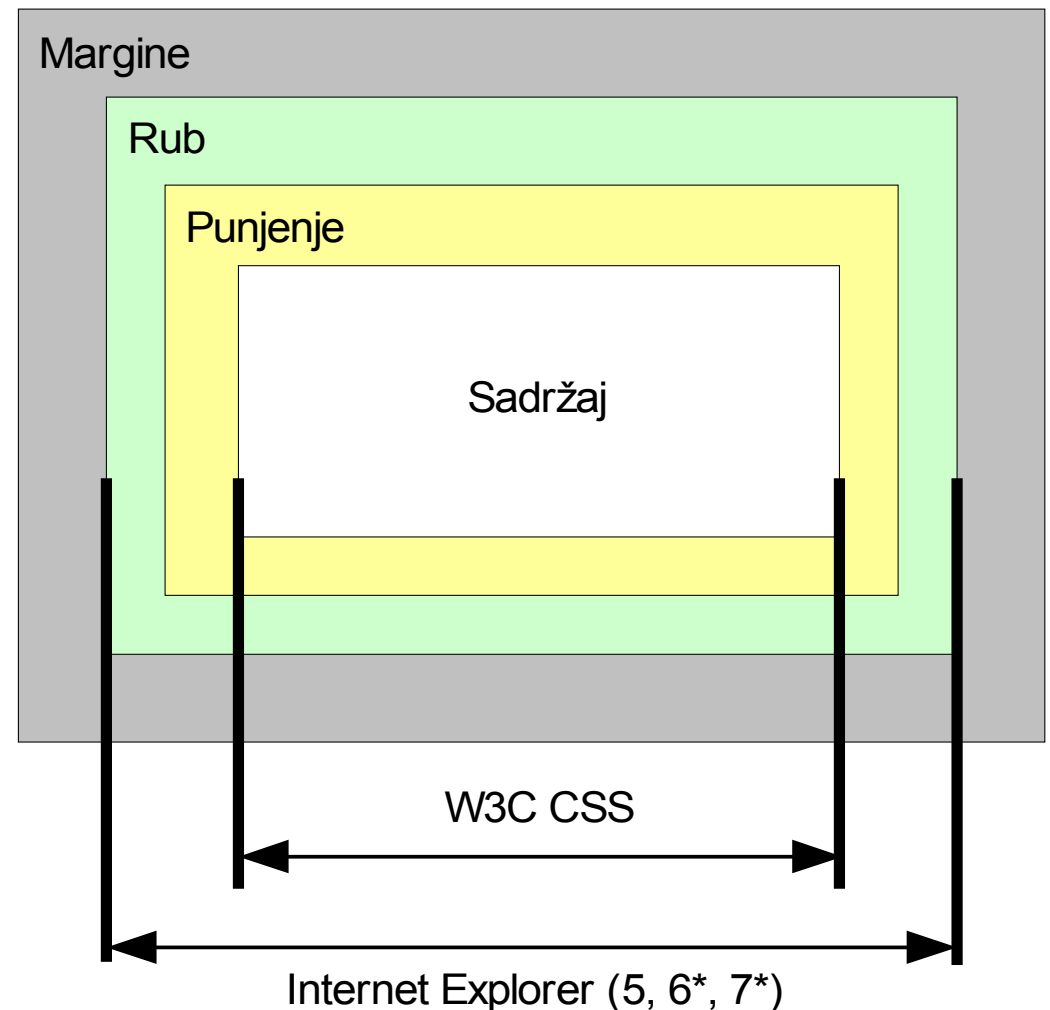


Izvor: libpng testovi,, <http://www.libpng.org/pub/png/pngs-img.html>

- Prvi red: IE6/Win
- Drugi red: IE7, Firefox 2
- Slika desno je 32-bitna (RGBA, transparentcija s alfa-kanalom)

Primjer: CSS

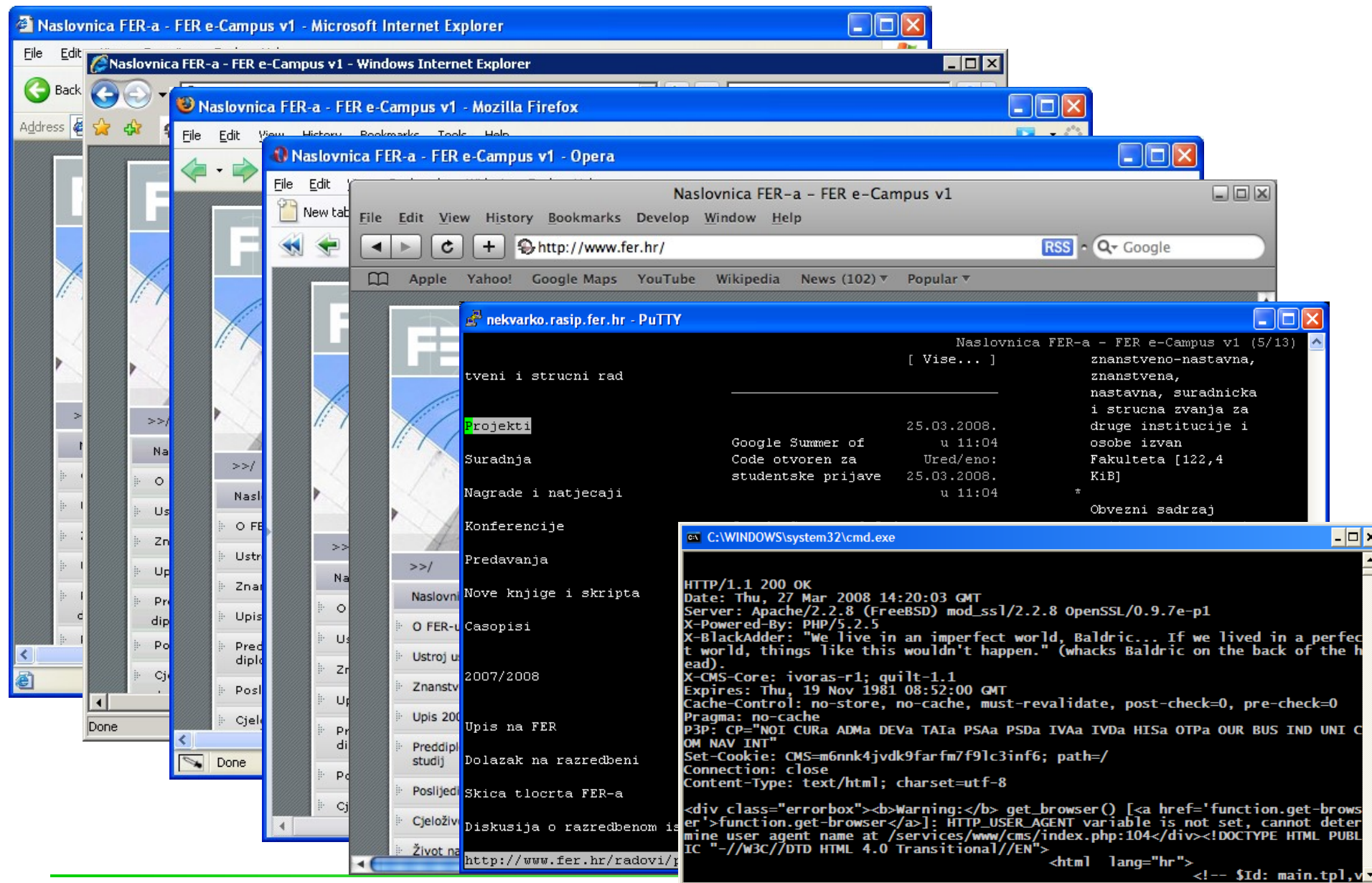
- Veličina elementa u CSS-u definirana standardom
- Element može imati
 - Sadržaj
 - Punjenje (padding)
 - Rub (border)
 - Margine (margin)



Primjer: CSS

- Internet Explorer pod određenim uvjetima koristi *svoj* model
- U obranu IE može se reći da je njegova interpretacija bliža stvarnom svijetu (kutija i sadržaj)
- Dodatna *mogućnost izbora* – **način rada** preglednika
- Način rada preglednika ovisi o dokumentu koji prikazuje
 - Deklaracije u zaglavlju dokumenta određuju kako će se preglednik ponašati

Primjeri preglednika



Sučelje Web aplikacije

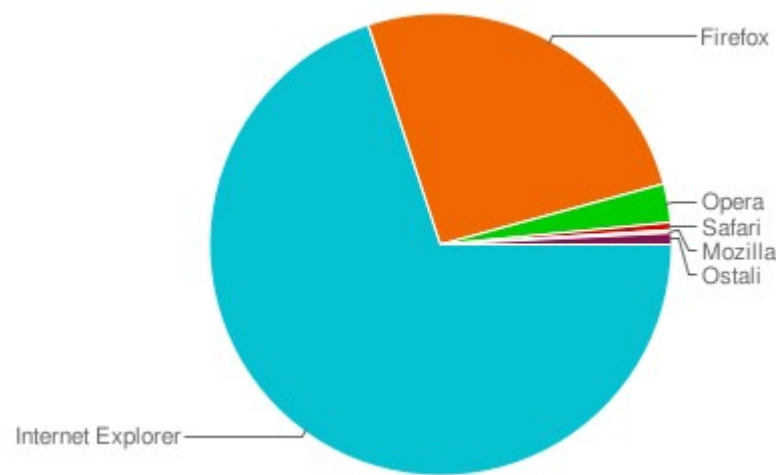
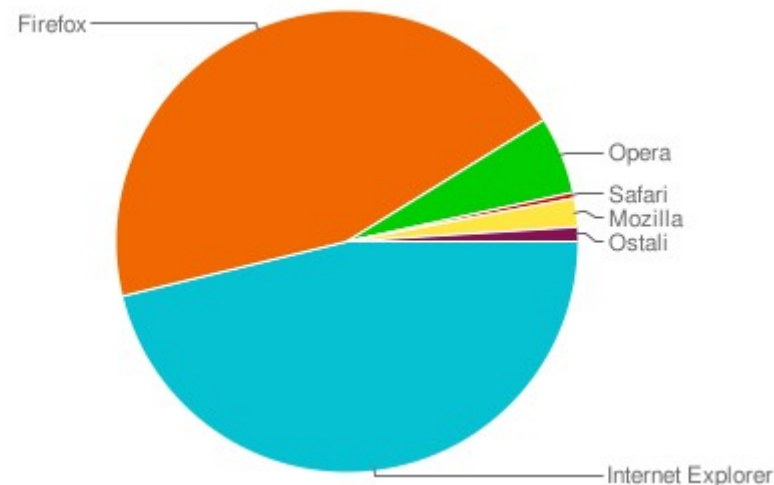
- Aplikacija mora imati dosljedno sučelje
 - Korisnik očekuje (gotovo) istovjetno sučelje neovisno o tome koji *monitor* koristi
- Ista aplikacija u različitim preglednicima
 - Mora biti potpuno funkcionalna
 - Trebala bi izgledati podjednako, ako ne i istovjetno

Ratovi preglednika

- Borba za dominaciju na tržištu
 - Brzina, lakoća korištenja, mogućnosti
 - Iako su preglednici besplatni za korištenje, prevlast donosi tržišni utjecaj i indirektnu korist
- Korisnici dobivaju
 - Poštivanje novih normi, npr. CSS3
 - Proširenja u postojećim normama
 - Preglednici su u prednosti ako privuku korisnike novostima, npr. transparentnost slika u CSS-u
 - `filter: alpha(opacity=50);` za IE
 - `-moz-opacity: 0.5;` za Mozillu
 - `opacity: 0.5;` po standardu, radi u novijim browserima
 - Novosti u sučelju
 - Npr. provjera pravopisa u Firefoxu

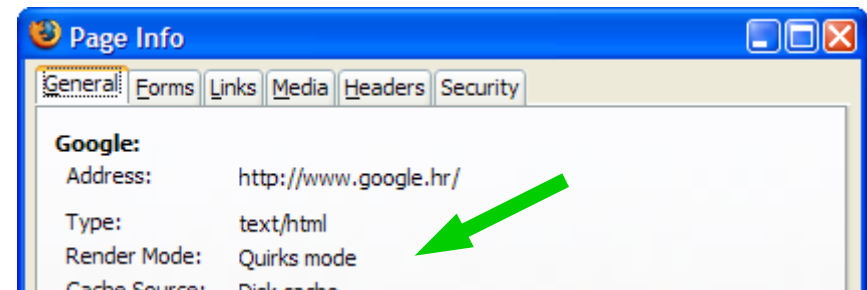
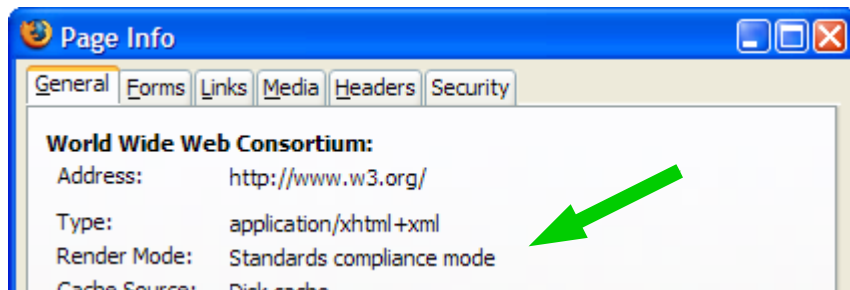
Zastupljenost preglednika

- Koji preglednik podržati?
- Statistika za dva portala za prvu polovicu ožujka 2008.
 - Gore: naslovnica weba FER-a
 - Dolje: naslovnica zdravstvenog portala Cybermed
- Zastupljenost Firefoxa u široj publici polako raste
- Izabrati vodećeg? Vodeću dvojku? Trojku?



Zastupljenost preglednika

- Koji preglednik podržati?
- Odgovor: pitanje je pogrešno postavljeno
- Pravo pitanje: **Koju normu izabrati?** (HTML X.X?)
- Pisanjem dokumenata možemo odrediti **način rada** preglednika
 - Poštivanje normi (*standards mode*)
 - Podrška vlastitim neobičnostima (*quirks mode*)



Odabir načina rada preglednika

- Ovisi o pregledniku :)
- Puna podrška normama (*standards mode*)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "  

http://www.w3.org/TR/html4/strict.dtd>
```
- Podrška normama uz određene *labavosti* (*almost standards mode*)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  

  Transitional//EN" "  

http://www.w3.org/TR/html4/loose.dtd>
```
- Poštivanje neobičnosti (*quirks mode*)
 - Bez ili s nepotpunim DOCTYPE-om (npr. Nedostaje URI)

The background of the slide features large, light green, stylized letters 'O' and 'R' that are partially visible and overlap each other. The 'O' is on the left and the 'R' is on the right, with a vertical white line separating them.

Protokol HTTP

HTTP



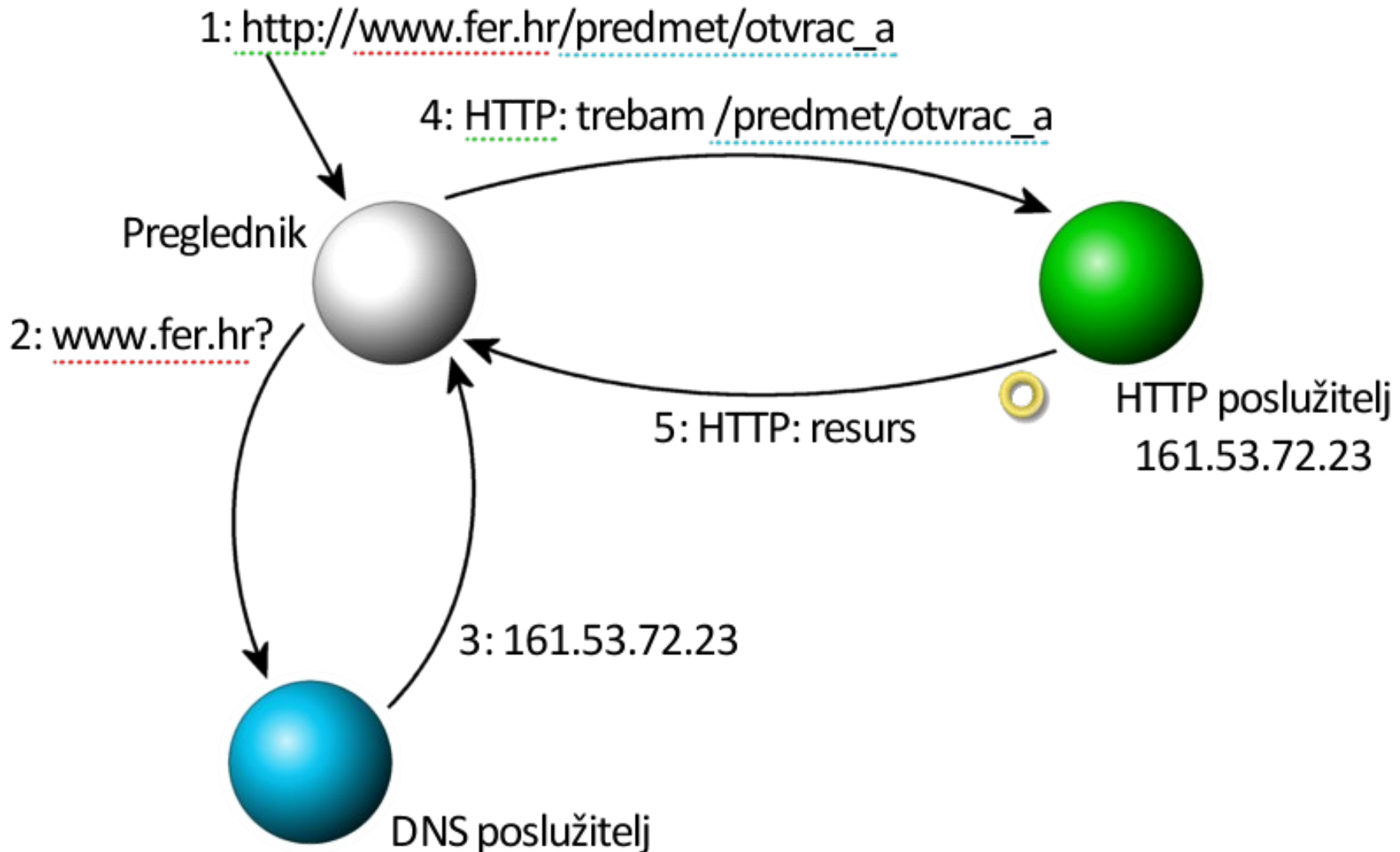
- Hypertext Transfer Protocol
 - HTTP 0.9 – 1991. (kasnije dokumentiran u RFC 1945)
 - HTTP 1.0 – 1996. (RFC 1945)
 - HTTP 1.1 – 1997. (RFC 2068) -1999. (RFC 2616)
- U uporabi uglavnom HTTP 1.1
- HTTP 0.9 – inicijalna verzija, iznimno jednostavna
- HTTP 1.0 – prva široko upotrebljiva verzija
- HTTP 1.1 – dogradnja 1.0 – kontrola veze, cacheiranje, sigurnost, adresiranje, kontrola pogreške, ...

HTTP



- ~~Hypertext~~ Transfer Protocol
- Prenosi **resurse** – *Resource Transfer Protocol*
- Resurs
 - Obična datoteka (*file transfer*)
 - Podaci o datoteci, meta-podaci (*meta-data*)
 - Dio datoteke – odsječak (*chunk*)
 - Dodatni podaci zahtjeva (*query information*)
 - Višedjelni podaci (*multipart data*)
 - Rezultat obrade (*processing result*)
 - ... **bilo što** što možemo imenovati!

Prijenos resursa



Resursi

- URI – metoda imenovanja (adresiranja) resursa
- HTTP – metode manipulacije resursima
- Web – sustav za rad (stvaranje, dohvat, promjenu i brisanje) s resursima
 - Zasnovan na kombinaciji URI+HTTP
 - MIME – opis oblika resursa
- Osnovna arhitektura (stil):
 - REST – Representational State Transfer
 - O ovome kasnije

Osnovne karakteristike

- Arhitektura **klijent** – **poslužitelj** (*client – server*)
- Klijent, agent korisnika (*browser, user agent*) – preglednik Web
 - Komunicira s poslužiteljem
 - Prikazuje rezultate komunikacije (resurse) korisniku
- Poslužitelj – poslužitelj weba (*web server*)
 - Osluškuje klijente
 - Priprema resurse i šalje ih klijentima
- U naravi, klijent i poslužitelj su **procesi** na proizvoljnom **računalu** (istom ili odvojenom)

Osnovne karakteristike

- Komunikacija procesa klijenta i poslužitelja
 - Adresiranje: URI
 - Mehanizam komunikacije: *mreža* (nebitno)
 - Komunikacijski kanal: *mreža* (nebitno)
 - Protokol komunikacije: HTTP
- HTTP od komunikacijskog kanala očekuje
 - Pouzdanost
 - Osiguran pouzdan prijenos resursa kanalom
 - Kontrolu protoka
 - Dojavu u slučaju pogreške
- Najčešće: HTTP preko TCP-a (koristi se port 80)

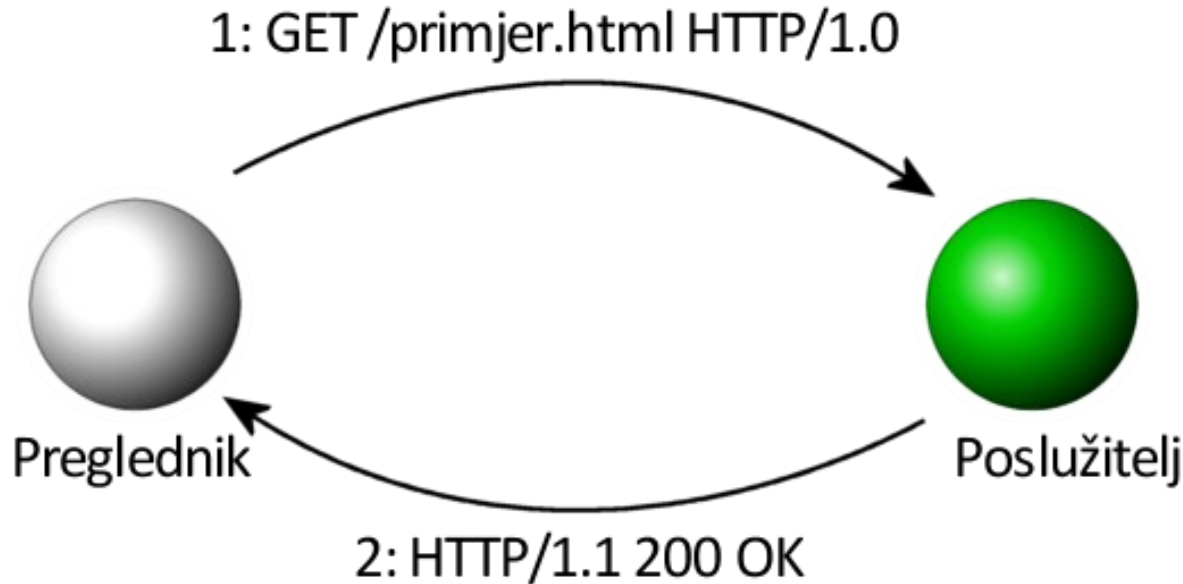
Konverzacija



- Vrlo jednostavna

GET /primjer.html HTTP/1.0

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.12) Gecko/20080201 Firefox/2.0.0.12



HTTP/1.1 200 OK

Date: Tue, 18 Mar 2008 00:56:40 GMT

Server: Apache/2.2.8 (FreeBSD)

Content-Type: text/html; charset=utf-8

<...sadržaj resursa...>

Osnovne karakteristike

- Princip: **zahtjev** – **odgovor** (*request – response*)
- Klijent šalje zahtjev (**poruku**) poslužitelju i očekuje odgovor (**poruku**)
- Poruke HTTP-a su **tekstovne**:

$$\begin{aligned} \text{generic-message} = & \text{start-line} \\ & *(message\text{-}header\ CRLF) \\ & CRLF \\ & [message\text{-}body] \end{aligned}$$
- Upravljački dio poruke (početni redak i zaglavlja) su čitljivi ljudima
- Tijelo poruke može i ne mora biti čitljivo ljudima

Osnovne karakteristike

- Tijelo poruke HTTP-a sadrži resurs(e)
- Tijelo poruke može sadržavati
 - Cjelokupan resurs
 - Dio, odsječak resursa (*chunking*)
 - Više resursa (*multipart*)
- Vrsta tijela poruke označena je **oznacom MIME tipa**
- Veličina tijela poruke – zapisana **unutar zaglavlja poruke**
 - Zaglavlje poruke navodi **broj okteta** tijela poruke (*octet counting*)

Osnovne karakteristike

- Ciklus zahtjev-odgovor = jedna **konverzacija**
- Poruka zahtjeva sadrži sve informacije potrebne za ispunjenje zahtjeva
- Slanjem odgovora konverzacija završava (ispunjen je zahtjev)
- Poslužitelj ne čuva stanje između dviju konverzacija s istim klijentom
- HTTP je protokol **bez očuvanja stanja** - *stateless*

Poruka HTTP-a



- Primjer jednostavne poruke zahtjeva

```
POST /primjer.html HTTP/1.1
Host: www.rasip.fer.hr
Content-Length: 21
Content-Type: text/plain
```

Ovo je tijelo poruke.

- Tijelo sadrži običan tekst (*text/plain*)
- Tijelo poruke duljine je 21 okteta

Struktura poruke

- Početni redak (*start-line*)
 - Zahtjev
Request-Line = Method Request-URI HTTP-Version CRLF
 - Odgovor
Response-Line = HTTP-Version Status-Code Reason-Phrase CRLF
- Zaglavlja (*message-headers*)
 - Ovisi o poruci
message-header = field-name ":" [field-value]
- Tijelo poruke (*message-body*)
 - Resurs kodiran na odgovarajući način kao **entitet**

Zahtjev

- Određuje **metodu** HTTP-a
 - Operacija nad resursom
- Određuje **URI** resursa
 - Jednoznačno određuje resurs
- Dodatna oznaka verzije protokola
 - Različiti zahtjevi na strukturu poruke
 - Kompatibilnost metoda prema starijim verzijama
- Primjer:

GET /primjer.html HTTP/1.1

Metoda URI Verzija protokola

Metode HTTP-a

- Norma definira metode
 - GET
 - HEAD
 - POST
 - PUT
 - DELETE
 - ...
- Usporedba sa SQLom:
 - *SELECT*: GET, HEAD
 - *UPDATE*: POST
 - *INSERT*: PUT
 - *DELETE*: DELETE

Metoda GET

- GET
 - Vraća sadržaj URI-ja

```
Microsoft Telnet> o www.fer.hr 80
Connecting To www.fer.hr...
C:GET / HTTP/1.0
C:
S:HTTP/1.1 200 OK
S:Date: Tue, 18 Mar 2008 00:56:40 GMT
S:Server: Apache/2.2.8 (FreeBSD) mod_ssl/2.2.8 OpenSSL/0.9.7e-p1
S:X-Powered-By: PHP/5.2.3
S:X-BlackAdder: George: "Honestly Blackadder, I'm sick of you treating me like I'm
S:some sort of thickey. Well it's not me
S: that's thick, it's you Mr. thickey Black thickey Adder thickey."
S:X-CMS-Core: ivoras-r1; quilt-1.1
S:Expires: Thu, 19 Nov 1981 08:52:00 GMT
S:Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
S:Pragma: no-cache
S:Set-Cookie: CMS=9jc4mmf1ths6mpi07f1rs2rn21; path=/
S:Connection: close
S:Content-Type: text/html; charset=utf-8
S:
S:<...sadržaj resursa...>
Connection to host lost.
Press any key to continue...
```

Metoda HEAD

- HEAD
 - Vraća opis (meta-podatke) URI-ja

C:HEAD / HTTP/1.0

C:

S:HTTP/1.1 200 OK

S>Date: Tue, 18 Mar 2008 00:49:46 GMT

S:Server: Apache/2.2.8 (FreeBSD) mod_ssl/2.2.8 OpenSSL/0.9.7e-p1

S:X-Powered-By: PHP/5.2.3

S:X-BlackAdder: "Everything is still the same. Lord Percy is still unemployed, your

S:animal is still not housetrained, and

S:nursie here is still a few sticks short of a bundle."

S:X-CMS-Core: ivoras-r1; quilt-1.1

S:Expires: Thu, 19 Nov 1981 08:52:00 GMT

S:Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

S:Pragma: no-cache

S:Set-Cookie: CMS=ntq7jljah69h0ucq5vvba27lm6; path=

S:Connection: close

S:Content-Type: text/html; charset=utf-8

S:

S:

Ostale metode

- **POST**
 - Šalje dodatne podatke postojećem resursu
 - Anotacija resursa
 - Slanje ili objava poruke
 - Slanje bloka podataka procesu za obradu
 - Širenje skupa (baze) podataka novim podatkom (*append*)
- **PUT**
 - Dodaje novi resurs na poslužitelj
- **DELETE**
 - Briše resurs sa poslužitelja
- **OPTIONS, TRACE, CONNECT**
 - Nisu izvedene u svim poslužiteljima

Odgovor



- Struktura:

Response-Line = HTTP-Version Status-Code Reason-Phrase CRLF

- Određuje **rezultat metode** HTTP-a

- Zadan numerički (kôd rezultata, *Status-Code*) i tekstualno (opis, *Reason-Phrase*)

- Oznaka verzije protokola

- Može ili ne mora biti ista kao verzija protokola zahtjeva

- Primjer:

HTTP/1.1 404 Not Found

Verzija protokola Kôd rezultata Opis rezultata

Kôd rezultata

- Troznamenkasti broj
- Grupe prema prvoj znamenci:
 - 1xx: Informativni (*Informational*) – Zahtjev primljen, nastavak rada
 - 2xx: Uspjeh (*Success*) – Zahtjev je uspješno primljen, protumačen i prihvaćen
 - 3xx: Preusmjeravanje (*Redirect*) – Za ispunjenje zahtjeva potrebne su daljnje akcije
 - 4xx: Greška klijenta (*Client Error*) – Zahtjev nije ispravno oblikovan ili se ne može ispuniti
 - 5xx: Greška poslužitelja (*Server Error*) – Poslužitelj nije uspio ispuniti inače ispravan zahtjev

Kôd rezultata – primjer

- 100: Continue
- 200: OK
- 301: Moved Permanently
- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error
- 501: Not Implemented
- 503: Service Unavailable
- 505: HTTP Version not supported

Zaglavlja poruka

- Informacije o poruci
 - Općenite: npr. *Date*
 - Opis zahtjeva: npr. *Host, Accept-Language, User-Agent*
 - Opis odgovora: npr. *Server*
 - Opis entiteta resursa: npr. *Content-Length*
- Zaglavlja definirana normom
 - Nepoznata zaglavlja se zanemaruju
 - Vlastita (nestandardna) zaglavlja koriste prefiks *X-*, npr. *X-CMS, X-VerzijaMogServera*

Primjer zaglavlja

```
C:HEAD / HTTP/1.0
C:
S:HTTP/1.1 200 OK
S:Date: Tue, 18 Mar 2008 00:49:46 GMT
S:Server: Apache/2.2.8 (FreeBSD) mod_ssl/2.2.8 OpenSSL/0.9.7e-p1
S:X-Powered-By: PHP/5.2.3
S:X-BlackAdder: "Everything is still the same. Lord Percy is still unemployed, your
S:animal is still not housetrained, and
S:nursie here is still a few sticks short of a bundle."
S:X-CMS-Core: ivoras-r1; quilt-1.1
S:Expires: Thu, 19 Nov 1981 08:52:00 GMT
S:Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
S:Pragma: no-cache
S:Set-Cookie: CMS=ntq7jljah69h0ucq5vvba27lm6; path=/
S:Connection: close
S:Content-Type: text/html; charset=utf-8
S:
S:
```

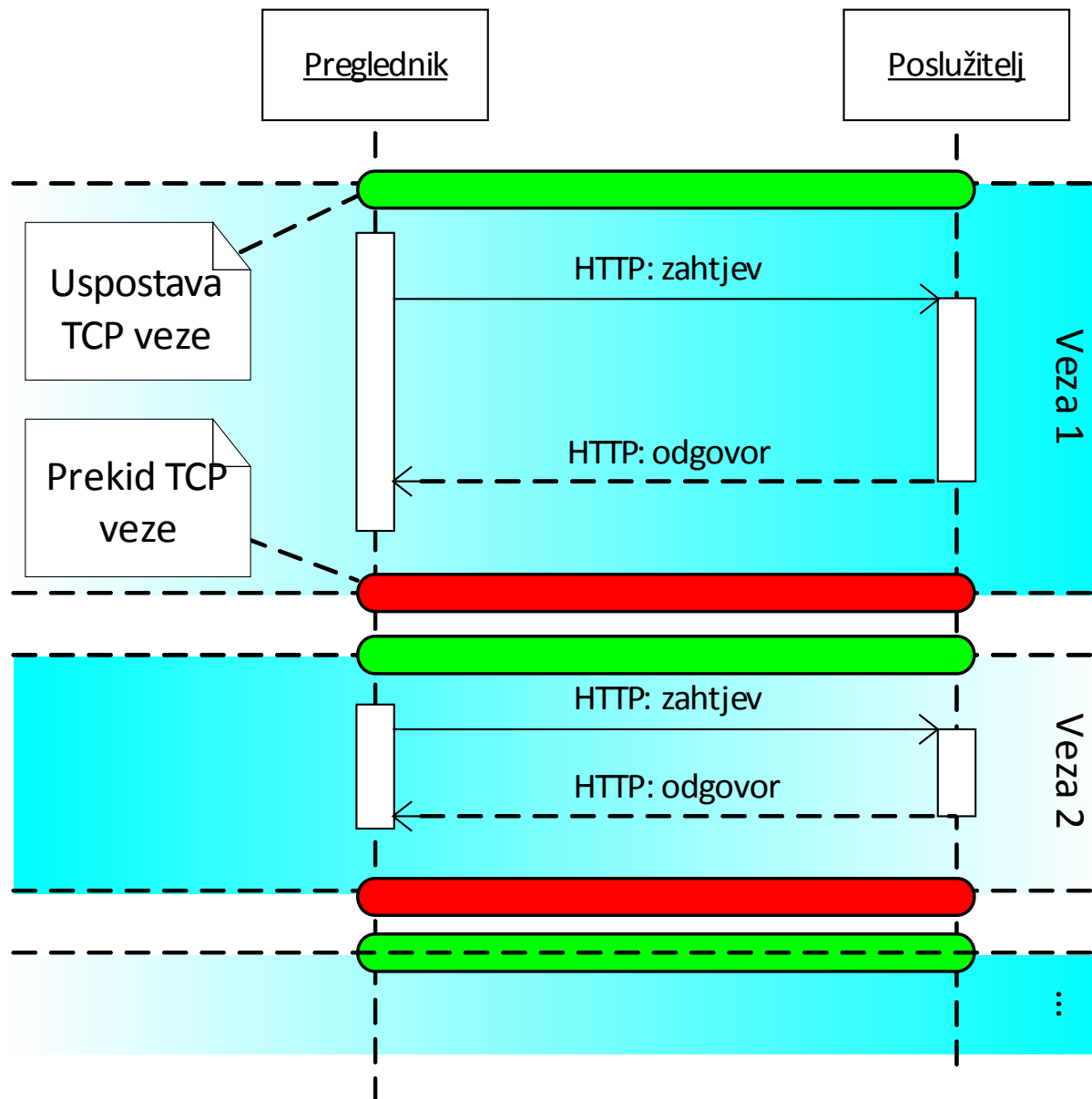
Upravljanje vezom

- Tipična stranica sadrži
 - 1 resurs – HTML
 - 1 resurs – CSS/JavaScript
 - 10ak resursa – slike
 - ... sve na **istom** poslužitelju
- Pristup svakom resursu je jedna **konverzacija** protokola
- HTTP je izgrađen nad mrežnim protokolom – konverzacije se odvijaju kroz mrežne **veze**

Upravljanje vezom – HTTP 1.0

- Nema promjena od HTTP 0.9
- Jednostavan pristup
- Jedna TCP veza po konverzaciji
 - Gubitak vremena na uspostavu veze
 - Zauzimanje resursa za uspostavljanje mnogo kratkotrajnih veza
 - Period prilagođavanja TCP-a stanju veze usporava početak konverzacije

Upravljanje vezom – HTTP 1.0



Upravljanje vezom – HTTP 1.1

- Najveće promjene u odnosu na HTTP 1.0
- Povećana stabilnost i sposobnost oporavka
- Mogućnost postupnog uvođenja novosti u protokol

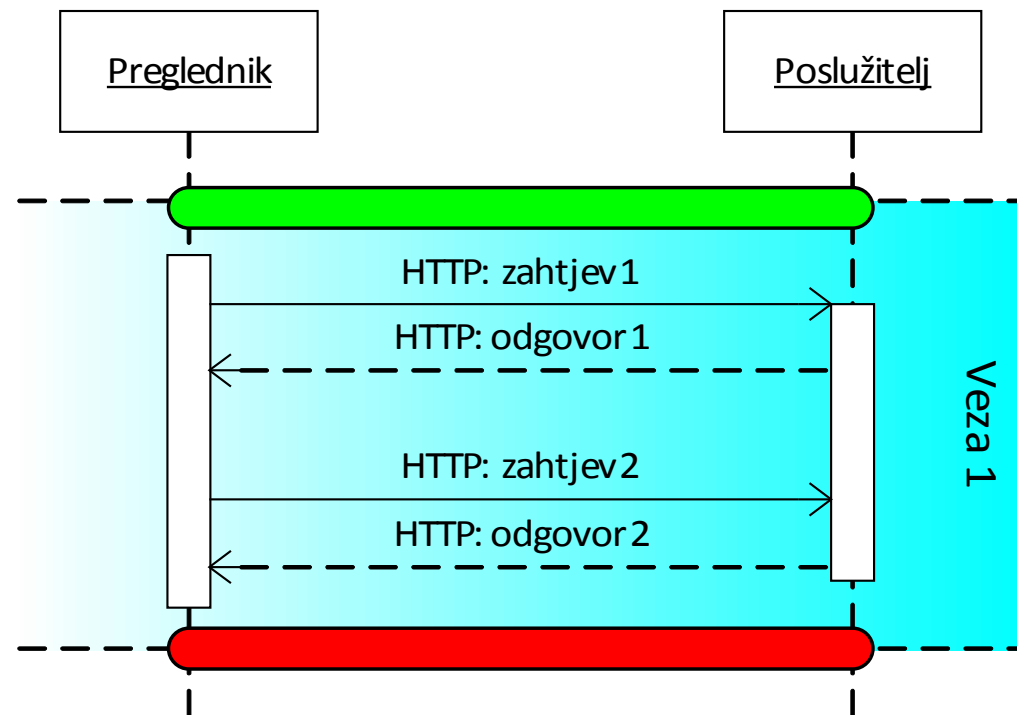
- HTTP 1.1 donosi
 - Trajne veze (*persistent connections*)
 - Cjevovod zahtjeva (*pipelined connections*)
 - Pregovaranje i upravljanje vezom

Trajne veze



- Trajne veze (*persistent connections*)

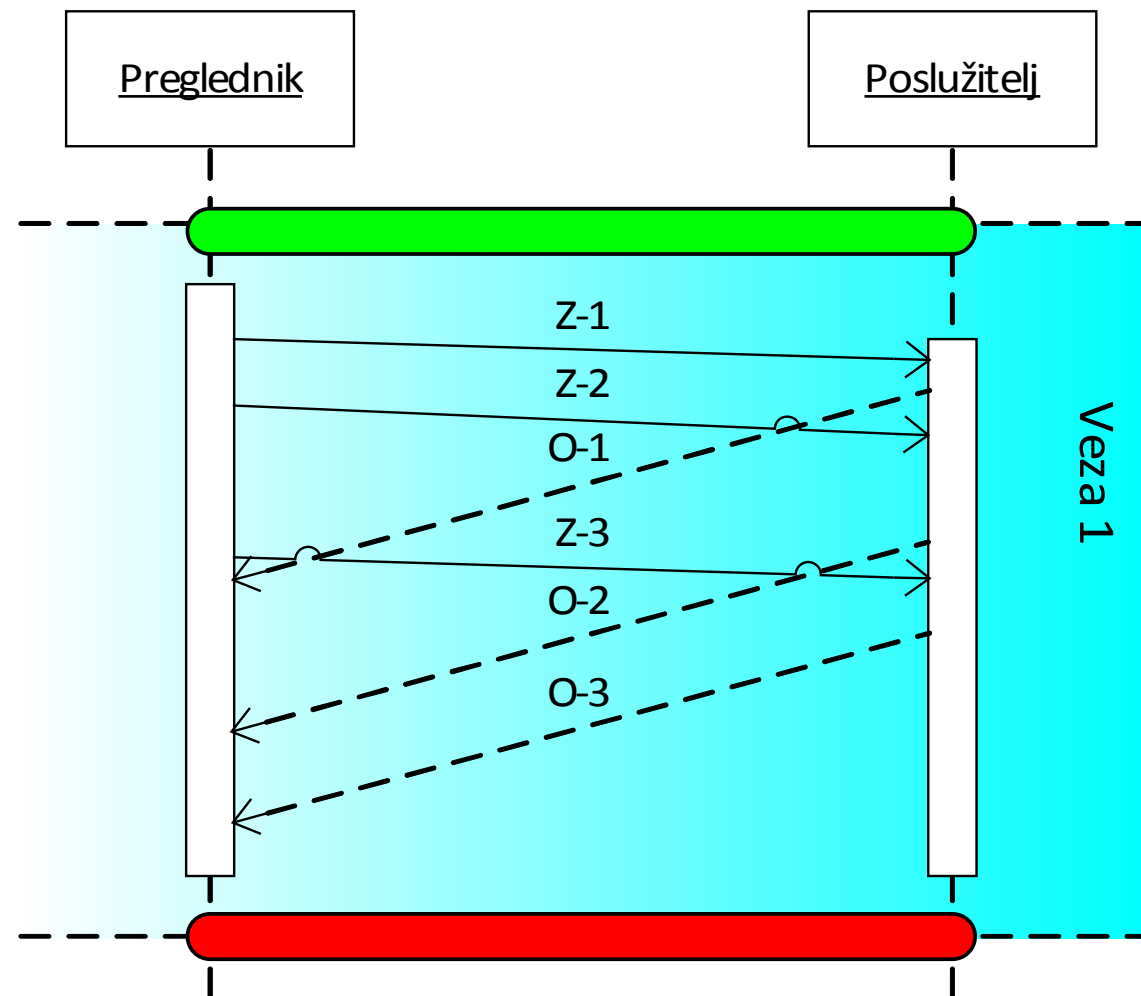
- Veza ostaje uspostavljena neko vrijeme (nekoliko sekundi)
- Za vrijeme trajanja veze može se obaviti nekoliko konverzacija
 - Ušteda vremena za uspostavu TCP veze
 - Ušteda resursa mrežne infrastrukture za održavanje većeg broja veza



Cjevovod zahtjeva

- Cjevovod zahtjeva (*pipelined connections*)

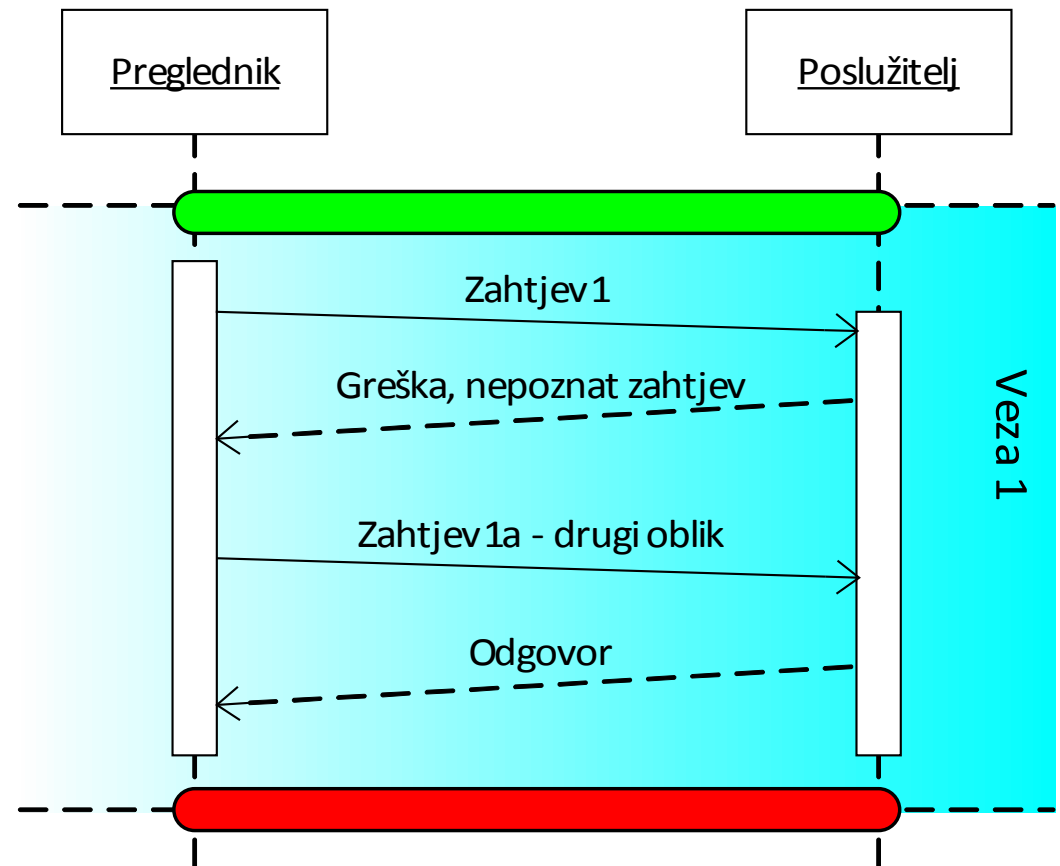
- Klijent šalje nekoliko zahtjeva u slijedu, ne čekajući odgovor
- Poslužitelj obrađuje zahtjeve kako stižu i šalje odgovore
- Redoslijed odgovora mora odgovarati redoslijedu zahtjeva



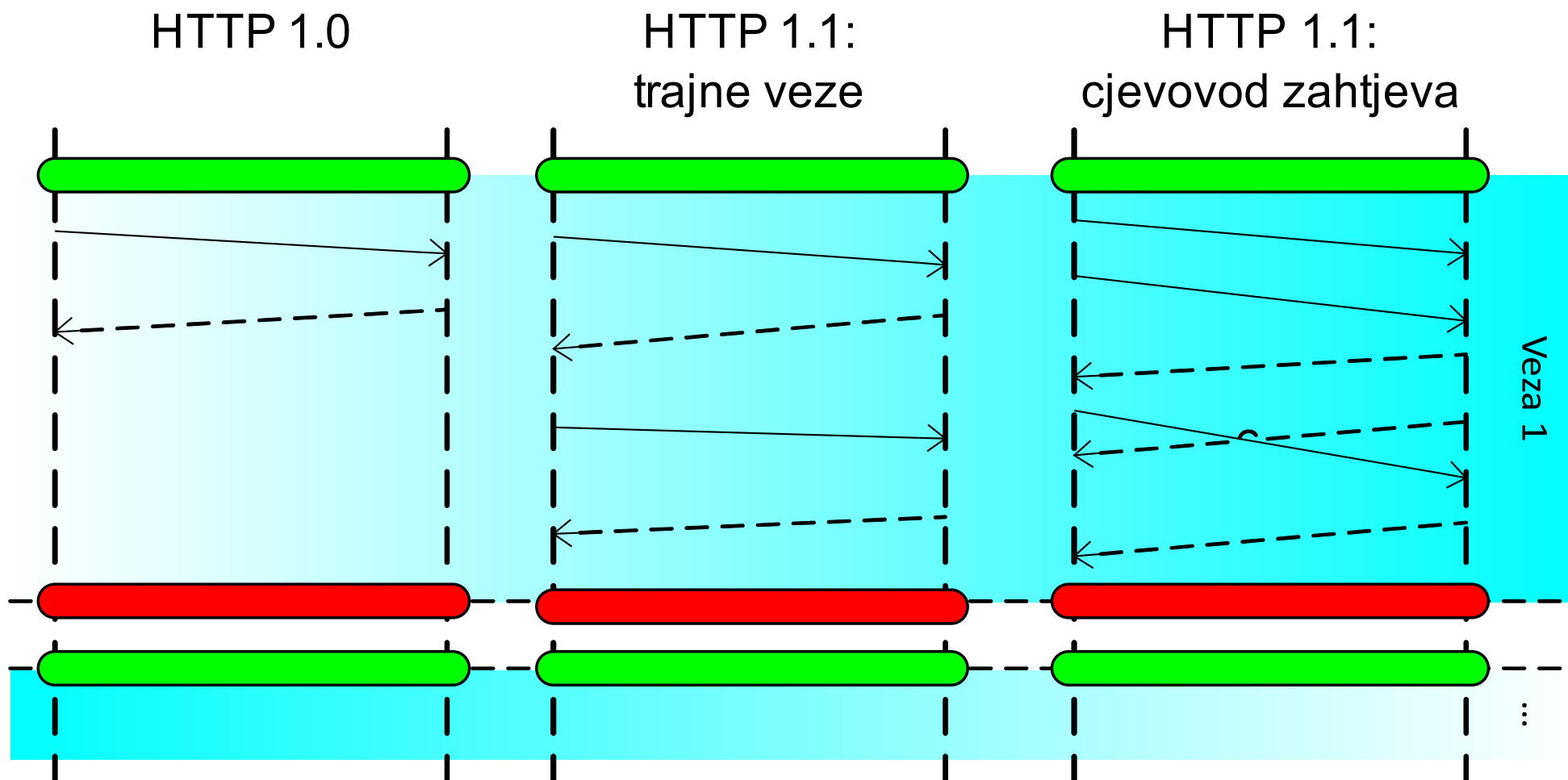
Pregovaranje i upravljanje vezom



- Pregovaranje i upravljanje vezom
 - Klijent ili poslužitelj mogu zahtijevati zatvaranje veze nakon jedne konverzacije (pretpostavlja se da se koriste trajne veze)
 - Klijent može provjeriti podržava li poslužitelj pojedine vrste zahtjeva i nastaviti rad bez prekida veze



Veze – HTTP 1.1



Primjer razlika verzija HTTP-a

- Minimalni zahtjev metodom GET
- HTTP 0.9 (nema zaglavlja ni verzije protokola)
GET /primjer.html
- HTTP 1.0 (verzija obvezna, zaglavlja nisu)
GET /primjer.html HTTP/1.0
- HTTP 1.1 (verzija i neka zaglavlja obvezna)
GET /primjer.html HTTP/1.1
Host: www.primjer.hr

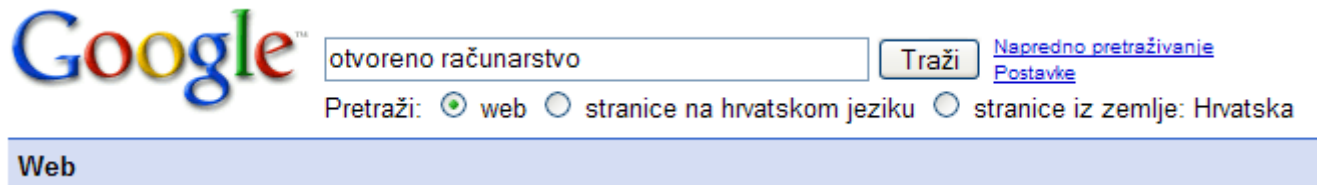
Zanimljiva zaglavlja

- Connection – kontrola veze
 - *Connection: close* – ako **sugovornik** ne podržava trajne veze
- Host – podrška za posrednike (*proxy*) i virtualne poslužitelje (*virtual host*)
 - Obavezno zaglavlje zahtjeva
 - Sadrži ime poslužitelja na kojeg se zahtjev odnosi
- Content-Type – oznaka MIME tipa entiteta poruke
 - Govori sugovorniku kako tumačiti resurs
- Content-Length – oznaka duljine sadržaja poruke
- Content-Range – oznaka dijela resursa na koji se poruka odnosi (npr. od 1000.-1546. okteta)

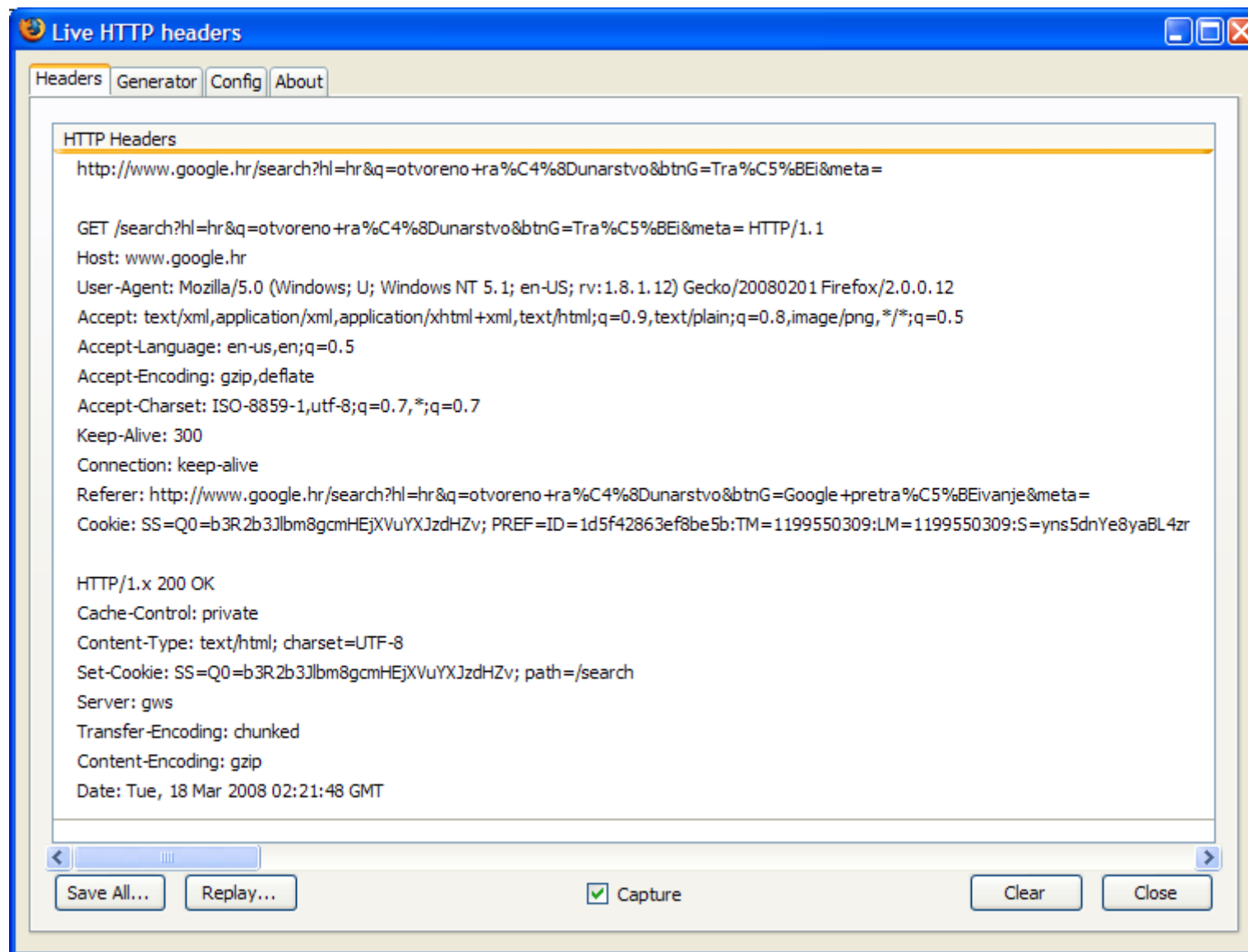
Primjer komunikacije



- Live HTTP headers
 - Dodatak na Firefox
 - Omogućuje praćenje, snimanje i ponovno reproduciranje komunikacije HTTP-om
- Traženje pojma na Googleu:



Primjer komunikacije



Primjer komunikacije 2

- Jednostavan obrazac

```
<html>
```

```
<form action="#" method="post">
```

```
  Pojam: <input type="text" name="pojam">
```

```
  <input type="submit" value="Pitaj">
```

```
</form>
```

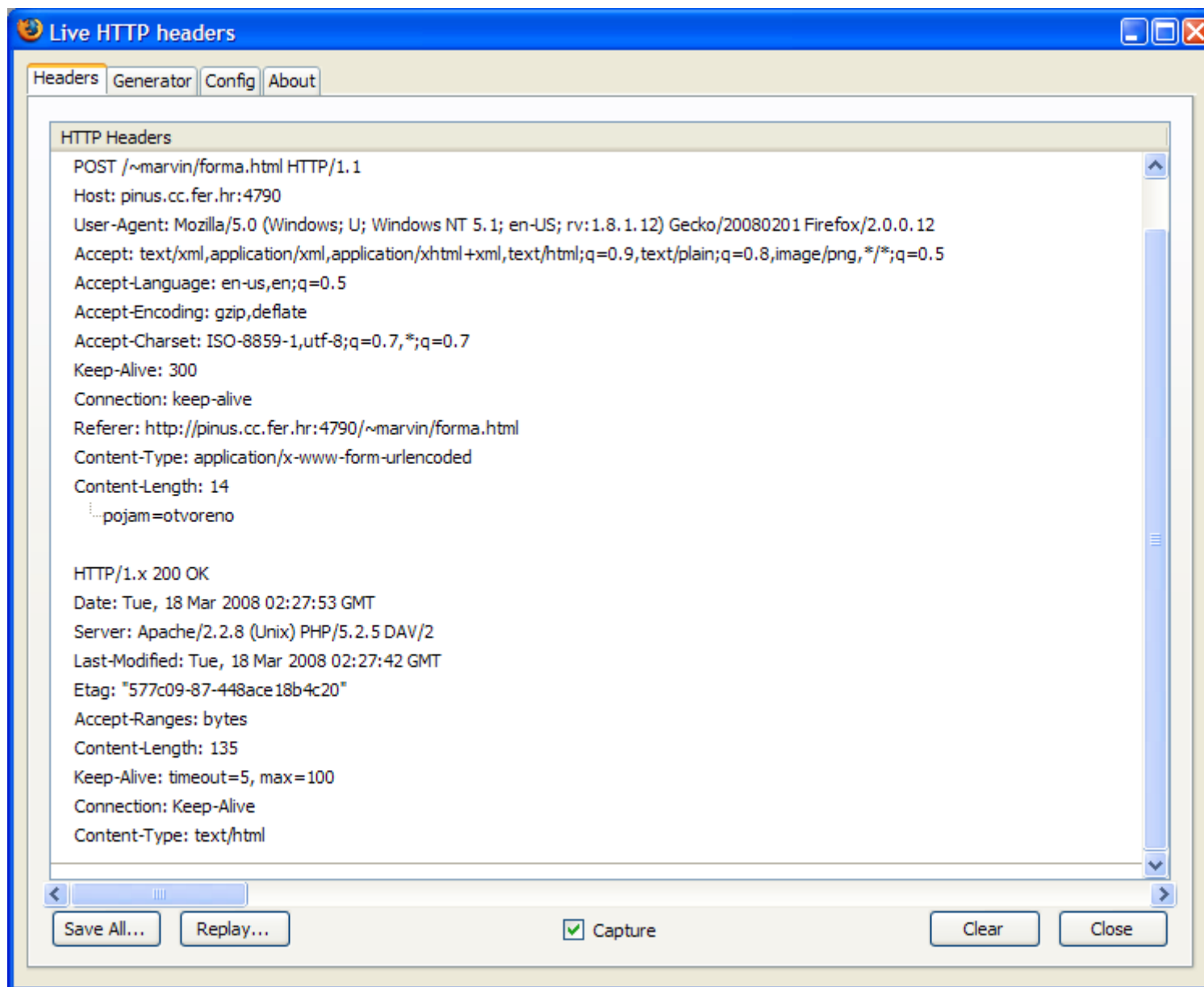
```
</html>
```

- Šalje podatke poslužitelju



Pojam:

Primjer komunikacije 2



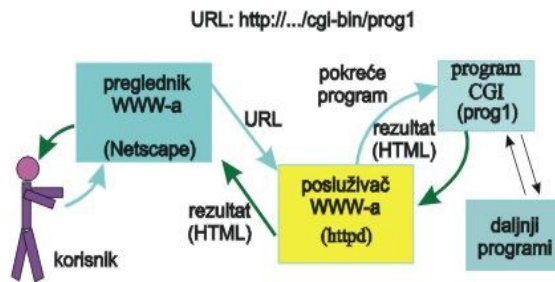


Poslužitelj - CGI

Uvod u CGI /1

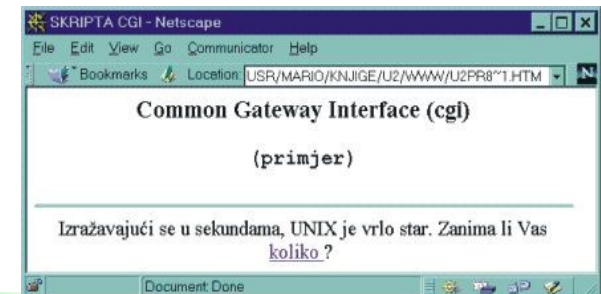


M.Žagar, UNIX i kako ga iskoristiti (<http://www.fer.hr/rasip/knjiznica>)



```
<HTML>
<HEAD>
<TITLE>SKRIPTA CGI</TITLE>
</HEAD>
<BODY>
<CENTER><H3>Common Gateway Interface (cgi)</H2>
<H4><SAMP>(primjer)</SAMP></H3>
<HR>
Izražavajući se u sekundama, UNIX je vrlo star. Zanima li Vas
<A HREF="http://www.rasip.fer.hr/cgi-bin/U2_starost_unixa_cgi"> koliko</A>?
</CENTER>
</BODY>
</HTML>
```

U2Pr8.html



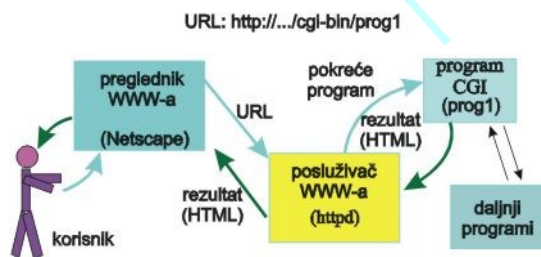
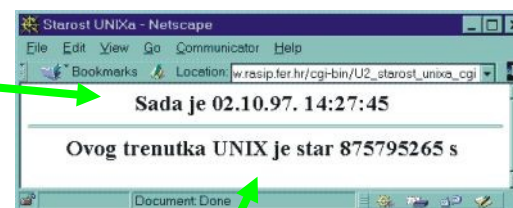
Uvod u CGI/2



M.Žagar, UNIX i kako ga iskoristiti (<http://www.fer.hr/rasip/knjiznica>)

```
U2_starost_unixa_cgi

#!/bin/csh
/bin/echo Content-type: text/html
/bin/echo
cat << KRAJ1
<HTML>
<HEAD>
<TITLE>Starost UNIXa</TITLE>
</HEAD>
<BODY>
<CENTER> <H3>
KRAJ1
echo Sada je ` /bin/date '+%d.%m.%y. %T' `
echo "<HR>"
U2_vrijeme_cgi
cat << KRAJ2
</H3> </CENTER>
</BODY>
</HTML>
KRAJ2
```



```
/* Starost UNIXa */
#include <stdio.h>
main ()
{
    long tim;
    time(&tim);
    printf("Ovog trenutka UNIX je star %d\n",tim);
}
```

U2_vrijeme_cgi

Web = sučelje

- Web može poslužiti kao sučelje prema aplikaciji
 - Korisniku se prikazuje sučelje opisano HTML-om
 - Korisnik pregledava, unosi i mijenja podatke u pregledniku
 - Na poslužitelju se pohranjuju i obrađuju podaci
- Kako izvesti vezu između sučelja i aplikacije?
 - Korisnik koristi preglednik za pristup aplikaciji
 - Preglednik komunicira s poslužiteljem mrežom
- Koji protokol koristiti?
- Kako prenijeti podatke sa sučelja?
- Kako promijeniti sučelje kao rezultat obrade?

Koji protokol koristiti?

- Preglednik je HTTP klijent
- Uvesti vlastiti protokol?
 - Potrebno implementirati svoj preglednik ili dodatak postojećem
 - Različite platforme?
 - Instalacija kod korisnika??
 - Održavanje preglednika ili dodatka???
- Rješenje: Koristimo **HTTP** kao protokol za komunikaciju između sučelja i ostatka aplikacije

Kako prenijeti podatke? I onda?



- HTML može opisati obrazac za unos podataka
- Podaci sa obrasca se **kodiraju** na klijentu i šalju protokolom HTTP **poslužitelju**...
- ...i što sad?
- Napisati **vlastiti** poslužitelj koji će znati obraditi podatke i vratiti rezultat pregledniku?
- Imamo **postojeću** aplikaciju, kako je integrirati u poslužitelj?

Common Gateway Interface

- Rješenje: Proširiti poslužitelj **mehanizmom** za prijenos podataka vanjskim programima – CGI
- Common **Gateway** Interface
 - Poslužitelj komunicira sa programima koji djeluju kao **poveznice** (*gateways*) weba sa ostalim sustavima
 - Zadržana općenitost i fleksibilnost (web ostaje web)
 - Nema intervencija u protokol HTTP
 - HTML ostaje jezik za opis sučelja prema korisniku
 - Poslužitelj HTTP-a ne mijenja se prema primjeni
 - Resurse (rezultate rada *poveznice*) preuzima poslužitelj weba i vraća korisniku (pregledniku)

Potrebni elementi

- Metode prijenosa podataka od preglednika do poslužitelja (HTTP-om)
- Metode prijenosa podataka od poslužitelja do poveznice
- Metode prijenosa rezultata od poveznice do poslužitelja
- Metode prijenosa rezultata od poslužitelja do preglednika (HTTP-om)

Preglednik – poslužitelj

- Procesi na različitim računalima povezani HTTP-om
- U okviru protokola HTTP možemo
 - Podatke poslati unutar URI-ja (dio upita, *query*)
foo://example.com:8042/over/there?name=ferret#nose
 - Podatke poslati unutar tijela poruke HTTP-a
generic-message = start-line
**(message-header CRLF)*
CRLF
[message-body]
- Podatke možemo poslati i u zaglavljima, ali time zadiremo u opis protokola HTTP

Poslužitelj – poveznica

- Procesi na istom računalu
- Možemo koristiti bilo koju metodu međuprocen-
ne komunikacije koju pruža operacijski sustav
 - Standardni ulaz / izlaz (*standard input/output*)
 - Datoteke (*files*)
 - Dijeljena memorija (*shared memory*)
 - Signali / događaji (*signals / events*)
 - Varijable okoline (*environment variables*)
 - Cjevovodi (*pipes*)
 - Utičnice (*sockets*)
 - Slanje poruka (*message passing*)
 - ...

Poslužitelj – poveznica

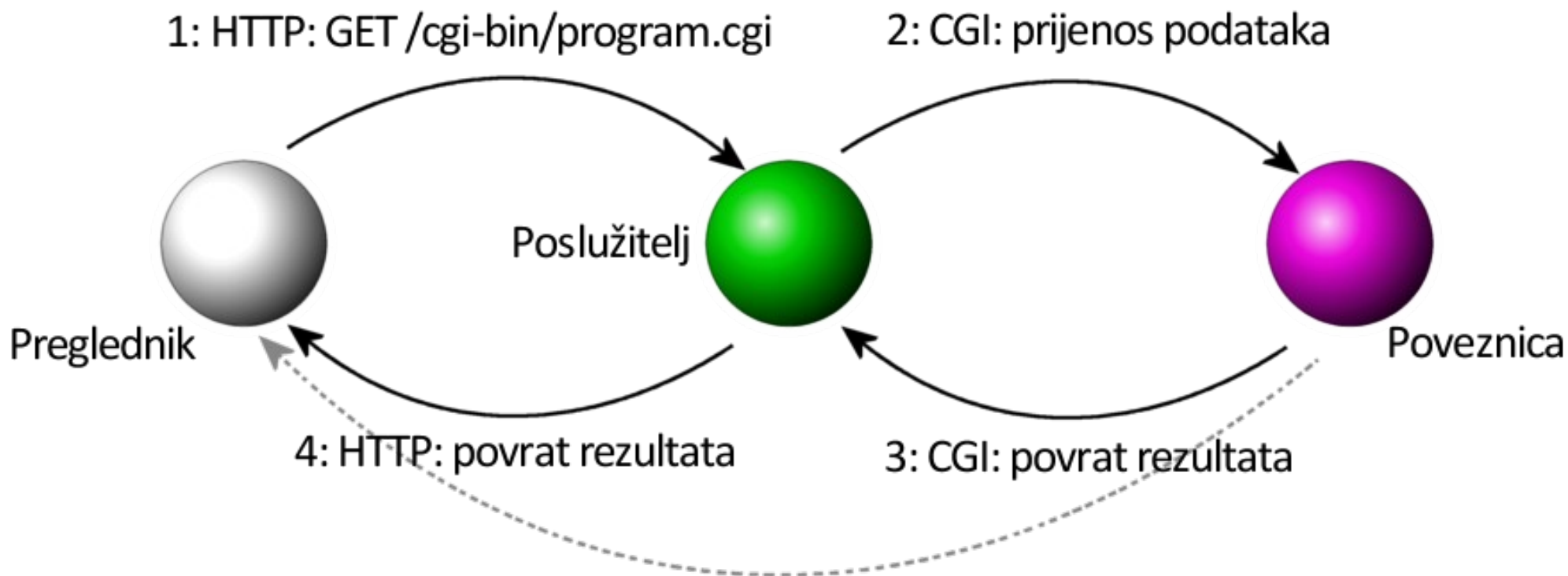
- Tražimo jednostavnu i brzu metodu podržanu na što više operacijskih sustava
 - Standardni ulaz / izlaz (*standard input/output*)
 - ~~Datoteke (*files*)~~
 - ~~Dijeljena memorija (*shared memory*)~~
 - ~~Signali / događaji (*signals / events*)~~
 - Varijable okoline (*environment variables*)
 - ~~Cjevovodi (*pipes*)~~
 - ~~Utičnice (*sockets*)~~
 - ~~Slanje poruka (*message passing*)~~
 - ...

Poveznica – poslužitelj

- Vrijedi isto kao i u obratnom smjeru
- Dodatni problem
 - Varijable okoline imaju ograničenu duljinu
- Koristimo standardni ulaz/izlaz
 - Nema ograničenja na količinu podataka koju poveznica može generirati
 - Izvedba u poslužitelju je gotovo jednaka kao čitanje obične datoteke

Poslužitelj – preglednik

- Prenosimo resurse
 - Nastali dinamički
 - Rezultat izvođenja poveznice
- Koristimo protokol HTTP



Što je CGI?



- CGI: Opis normiranog sučelja između **poslužitelja** (HTTP) i programa **poveznice**
- Mehanizam komunikacije između preglednika i poslužitelja u domeni je protokola HTTP
 - Nastavak opisa HTTP-a
 - Zaokružuje opis CGI
- Primjer korištenja – obrada podataka upisanih u obrasce opisane HTML-om na strani preglednika

Obrasci u HTML-u

- Jezik HTML može opisati obrasce (nisu dio CGI-a!)
- Obrasci se stvaraju oznakom <FORM ...> unutar koje se navode polja za unos - <INPUT ...> i dr.
- Opis obrasca - <FORM ...>
 - Određuje URI resursa kojem se šalju podaci iz obrasca (npr. adresa poveznice)
 - Određuje metodu kojom se podaci šalju (GET ili POST)
 - Određuje način kodiranja podataka (MIME tip)
 - Običan obrazac – "application/x-www-form-urlencoded"
 - Obrazac kojim šaljemo datoteku – "multipart/form-data"

Obrasci u HTML-u

- Metoda slanja podataka obrasca – GET ili POST
- GET
 - Za operacije koje nemaju utjecaja na stanje sugovornika (poslužitelja)
 - Npr. pretraživanje podataka (podaci se ne mijenjaju)
- POST
 - Za operacije koje mijenjaju stanje sugovornika (poslužitelja)
 - Za slanje datoteka (poljem `<INPUT type="file" ...>`)
 - Npr. brisanje ili uređivanje zapisa u imeniku (podaci se mijenjaju)

Polja za unos

- Polja moraju imati atribut imena (*name*) i vrijednosti (*value*)
- Prilikom slanja podataka (*submit*), stvara se popis parova (ime, vrijednost)
- Iznimka je polje `<INPUT type="image" ...>`
 - Klikom na sliku inicira se slanje podataka
 - Polje stvara 2 dodatna podatka – x i y koordinatu točke slike na koju je korisnik kliknuo mišem

`<INPUT type="image" name="boja" ... >`

podaci: boja, boja.x, boja.y

Primjer



- Prazan obrazac

Ime:

Prezime:

Spol: muški ☐ ženski ☐

Želim sudjelovati u anonimnoj anketi ☒

Moje raspoloženje:

```
<form action="obrada.cgi" method="post">
```

```
<input name="korisnik" value="145">
```

Ime:

Prezime:

Spol: muški ☐ ženski ☐

Želim sudjelovati u anonimnoj anketi ☒

Moje raspoloženje:

- Popunimo neka polja (obrišemo *checkbox*)

Ime:

Prezime:

Spol: muški ☒ ženski ☐

Želim sudjelovati u anonimnoj anketi ☐

Moje raspoloženje:

Prikupljanje podataka

- Skup podataka
 - korisnik: 145
 - ime: "moje ime"
 - prezime: "moje prezime"
 - spol: "m"
 - anketa: \emptyset
 - raspoloženje: \emptyset

```

<form action="obrada.cgi" method="post">
  <input name="korisnik" type="text" value="145">
  Ime: <input name="ime" size="32" type="text" value="moje ime">
  Prezime: <input name="prezime" size="32" type="text" value="moje prezime">
  Spol: muški <input name="spol" value="m" type="radio" checked="">
        ženski <input name="spol" value="f" type="radio">
  Želim sudjelovati u anonimnoj anketi <input name="anketa" value="da" type="checkbox">
  Moje raspoloženje: <input name="raspolozenje" type="range" value="0">
  <input type="button" value="Pošalji">

```

Kodiranje podataka

- Podaci se kodiraju za prijenos HTTP-om
- Pretpostavljeni MIME tip kodiranih podataka je 'application/x-www-form-urlencoded'
- Imena i vrijednosti se normiraju
 - Znakovi razmaka zamjenjuju se sa '+'
 - Posebni znakovi kodiraju se kao za URI
 - Ne-alfanumerički znakovi zamjenjuju se sa %HH (HH - heksadekadski zapis okteta znaka)
 - Znakovi novog reda predstavljaju se parovima CR LF, tj. '%0D %0A'

Kodiranje podataka

- Podaci se zapisuju kako se pojavljuju u dokumentu u obliku 'ime=vrijednost'
- Podaci se međusobno odjeljuju znakom '&'
- Podaci koji nemaju postavljenu vrijednost mogu se izostaviti
 - Neoznačene mogućnosti (npr. *radio* i *checkbox* dugmad) mogu se izostaviti u kodiranim podacima
- Primjer:
naziv=Otvoreno+ra%C4%8Dunarstvo&ects=4
(UTF-8 kôd znaka č je 0xC48D)

Kodiranje podataka

```
<form action="obrada.cgi" method="post">
```

Ime:

Prezime:

Spol: muški ☒ ženski ☐

Želim sudjelovati u anonimnoj anketi ☐

Moje raspoloženje:

- Skup podataka
 - korisnik: 145
 - ime: "moje ime"
 - prezime: "moje prezime"
 - spol: "m"
 - anketa: \emptyset
 - raspoloženje: \emptyset

- Kodiranje pojedinih podataka
 - "korisnik=145"
 - "ime=moje+ime"
 - "prezime=moje+prezime"
 - "spol=m"
- Kodiranje skupa podataka

"korisnik=145&ime=moje+ime&prezime=moje+prezime&spol=m"

Prijenos podataka poslužitelju

- Podaci se prenose porukom HTTP-a
- Kodirani skup podataka (*query string*) može se zapisati
 - Unutar URI-ja
 - Unutar tijela poruke HTTP-a
- Podaci zapisani unutar tijela poruke HTTP-a moraju biti označeni odgovarajućom oznakom MIME tipa
 - Poruka sadrži zaglavlje Content-Type, npr.

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

Primjer: metoda GET

- Podaci su upisani u URI

```
GET /~marvin/obrada.cgi?korisnik=145&ime=moje+ime&prezime=moje+prezime&spol=m HTTP/1.1
Host: pinus.cc.fer.hr:4790
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.12) Gecko/20080201
Firefox/2.0.0.12
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://pinus.cc.fer.hr:4790/~marvin/forma.html
```

- Tijelo poruke je prazno

Primjer: metoda POST

- Podaci su upisani u tijelo poruke
 - Content-Type: application/x-www-form-urlencoded
 - Content-Length: 53

POST /~marvin/obrada.cgi HTTP/1.1

Host: pinus.cc.fer.hr:4790

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.12)

Gecko/20080201 Firefox/2.0.0.12

Accept: text/xml,application/xml,application/xhtml+xml,

text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referer: http://pinus.cc.fer.hr:4790/~marvin/forma.html

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

Content-Length: 53

korisnik=145&ime=moje+ime&prezime=moje+prezime&spol=m

CGI – prijenos podataka poveznici



- Poslužitelj je primio poruku HTTP-a koja sadrži podatke u URI-u (iza znaka '?')
 - Kopira upit (*query string*) u varijablu okoline QUERY_STRING
 - Pokreće proces poveznice
 - Čeka kraj izvođenja poveznice
- Poslužitelj je primio poruku HTTP-a koja ima sadržaj
 - Pokreće proces poveznice
 - Prenosi sadržaj poruke poveznici preko standardnog ulaza
 - Čeka kraj izvođenja poveznice

Varijable okoline

- Prije pokretanja procesa poveznice poslužitelj postavlja varijable okoline koje sadrže dodatne podatke o poslužitelju, zahtjevu HTTP-a, ...
- Neke varijable okoline
 - QUERY_STRING – sadrži niz znakova upita
 - REQUEST_METHOD – sadrži HTTP metodu upita (GET, POST, ...)
 - CONTENT_LENGTH – sadrži duljinu niza znakova upita
 - REMOTE_ADDR – sadrži IP adresu korisničkog računala s kojeg je upućen zahtjev

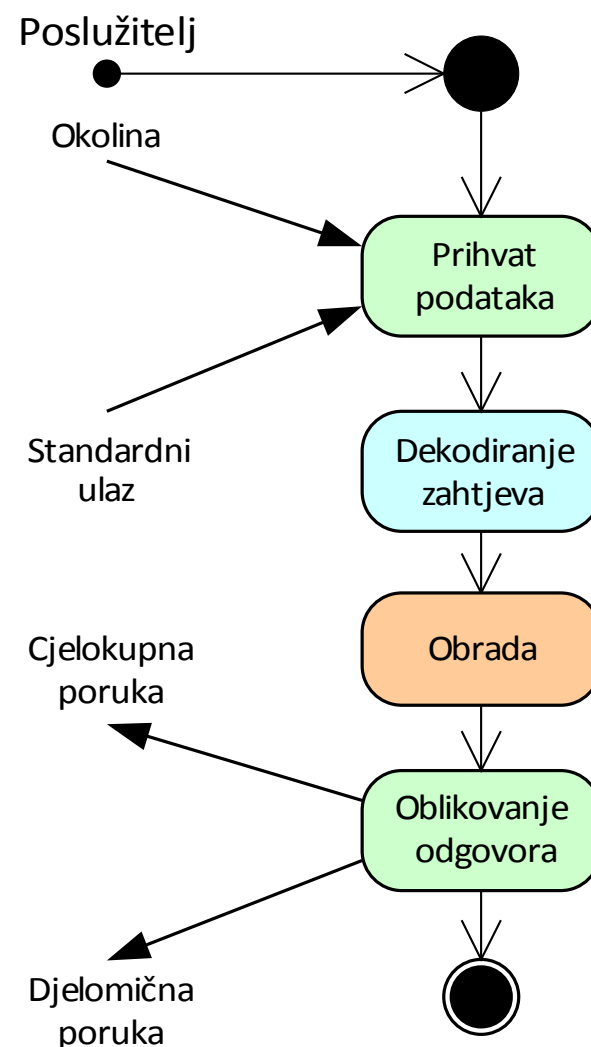
Pokretanje poveznice

- Poslužitelj pokreće poveznicu kao samostalan proces
- Poslužitelj preusmjeruje standardni ulaz i izlaz poveznice tako da mogu komunicirati
- Nakon pokretanja poveznice poslužitelj čeka kraj izvođenja
- Podatke primljene preko standardnog izlaza poveznice poslužitelj šalje HTTP-om kao odgovor na zahtjev

Rad poveznice



- Poveznica prihvaća podatke zahtjeva (query string)
 - Iz varijable okoline ili
 - Sa standardnog ulaza (stdin)
- Dekodira podatke zahtjeva (!)
- Obrada...
- Oblikuje rezultat (odgovor)
 - Cjelokupna poruka HTTP-a
 - Početni redak
 - Zaglavlja
 - Tijelo poruke
 - Djelomična poruka HTTP-a
 - Zaglavlja
 - Tijelo poruke

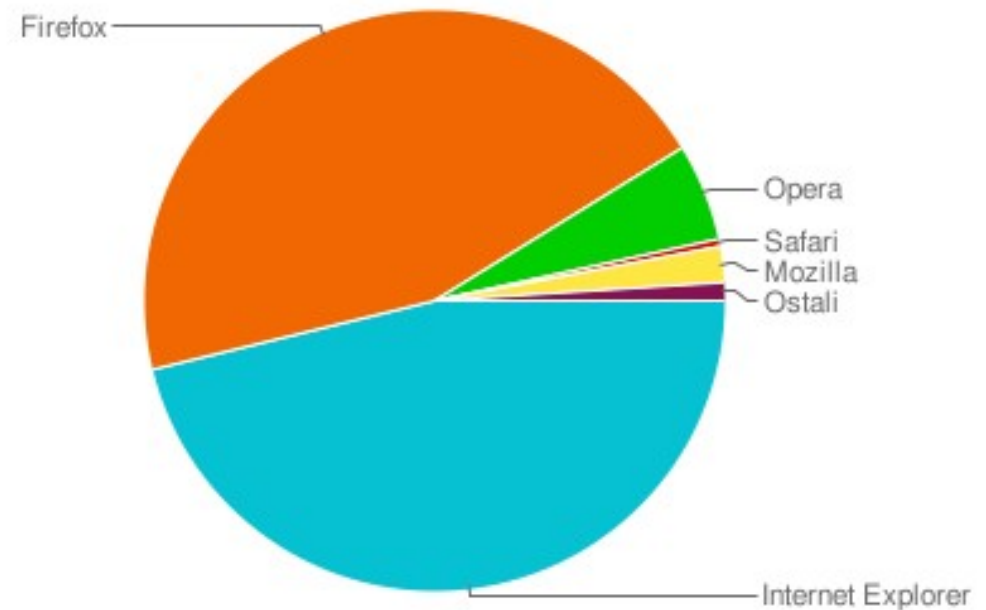


Povrat rezultata

- Rezultat rada poveznice je **resurs**
 - HTML dokument
 - XML dokument
 - Slika
 - Zvuk
 - ...
- Poveznica treba postaviti oznaku MIME tipa
 - Poslužitelj može postaviti pretpostavljenu (npr. text/html)
 - Preglednik ne može znati što je primio
- Oblik rezultata mora poštivati HTTP
 - Npr. prazan redak između zaglavlja i tijela poruke

Primjer

- Rezultat rada poveznice može biti i slika
- Primjer je [ovdje](#) (Google Chart API)
 - Content-Type: image/png



<http://chart.apis.google.com/chart?chs=330x150&cht=p&chl=Internet%20Explorer|Firefox|Opera|Safari|Mozilla|Ostali&chd=t:46.2,45,5.4,0.4,2,1&chco=05C1D1,F16702,01cd01,DD0000,FFE743,881155>

CGI - problemi

- Pokretanje poveznice je pokretanje novog procesa
- Za svaki zahtjev koji stigne HTTP-om treba
 - Pripremiti okolinu procesa
 - Pokrenuti proces
 - *Težak* posao OS-a
 - Čekati dok proces ne završi
 - Dva aktivna procesa, poslužitelja i poveznice
- Poveznice nisu pogodne za velika opterećenja
- Osnovni koncepti su široko primjenjivi
 - Proširenje mogućnosti poslužitelja
 - Osnova drugih tehnologija

CGI = suradnja

- Za rad poveznice potrebno je uskladiti
 - HTML – za opis sučelja prema korisniku
 - HTTP – za prijenos podataka (resursa) od i do korisnika
 - Poslužitelj HTTP – podesiti ga da prepozna je poveznice
 - Nepoznati operacijski sustav
 - Pokreće procese poslužitelja i poveznice
 - Omogućuje komunikaciju procesa
- Na ovim osnovama izgrađen je **mehanizam** prijenosa podataka između poslužitelja i vanjskih programa (poveznica) – CGI

