



**Preddiplomski studij  
Računarstvo**

# **Višemedijske usluge**

Upute za izvođenje 2. laboratorijske vježbe

**World Wide Web**

**Ak. god. 2007./2008.**

## Sadržaj

1	Uvod.....	2
2	Osnove kodiranja HTML dokumenata .....	2
2.1	Primjeri i zadaci.....	5
3	Osnove kodiranja HTML obrazaca .....	5
3.1	Primjeri i zadaci.....	7
4	Osnove skripti na poslužitelju.....	7
4.1	Primjeri i zadaci.....	9
5	Osnove skripti na klijentu .....	9
6	Zadaci.....	10
6.1	Pokretanje mrežne okoline .....	11
6.2	Kreiranje statičkog HTML dokumenta.....	14
6.3	Kreiranje HTML obrasca.....	18
7	Prebacivanje rješenja na Moodle.....	19
8	Korisni izvori .....	20

## 1 Uvod

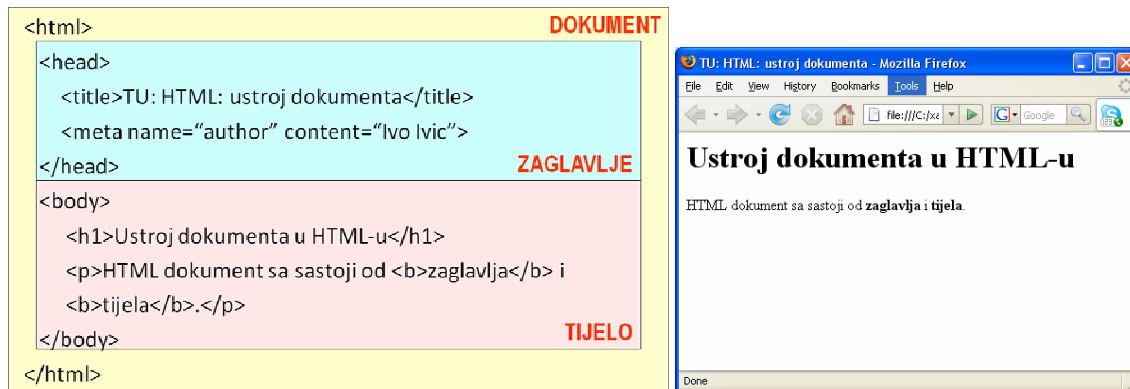
Cilj ove vježbe je upoznavanje s odabranim tehnologijama World Wide Web, kroz primjere i samostalan rad.

Za ovu vježbu koristit će se nekoliko alata:

- *VMware Player*, virtualizacijska okolina;
- *IMUNES*, mrežni simulator/emulator;
- *Wireshark* (prije *ethereal*), softverski protokolni analizator;
- *NEdit*, program za uređivanje teksta;
- *lighttpd*, web poslužitelj; i
- *Opera*, web preglednik.

## 2 Osnove kodiranja HTML dokumenata

Hypertext Markup Language (HTML) je osnovni format zapisa na webu. Svojstvo HTML-a kao jezika za označavanje (engl. *markup*) očituje se umetanjem oznaka (engl. *tag*) u običan tekst, pri čemu oznake utječu na predočavanje teksta i služe za uvođenje hiperveza. Slika 1 na primjeru pokazuje ustroj HTML i izgled tog dokumenta u pregledniku.



Slika 1. Ustroj HTML dokumenta (primjer s predavanja)

Početak i kraj HTML dokumenta označeni su otvorenom (<html>) i zatvorenom (</html>) oznakom. Dokument sastoji se od tijela i zaglavljja. Unutar zaglavljja (<head>) dokumenta definiraju se:

- naslov (<title>), koji se obično vidi u statusnoj traci preglednika, i
- drugi metapodaci („podaci o podacima“) koji se ne vide u pregledniku, ali mogu poslužiti za kontrolu prikaza, indeksiranje i pretraživanje i slično.

Jedan od važnih metapodataka je način kodiranja stranice. Na primjer, u stranicama na hrvatskom jeziku za ispravan prikaz slova č, ć, đ, š i ž treba ispravno postaviti način kodiranja:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

ili ovako:

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />
```

Oznaka *charset* odnosi se na skup znakova (engl. *character set*) u dokumentu. u prvom primjeru koristi se skup znakova *utf-8*. Oznaka UTF-8 odnosi se na 8-bitni *UCS/Unicode Transformation Format*. Tim načinom može se kodirati bilo koji znak u Unicode standardu, uz zadržanu kompatibilnost sa skupom znakova ASCII. Zato se UTF-8 sve češće koristi i za druge internetske aplikacije (npr., e-mail) i zapis tekstualnih dokumenata. Drugi primjer (*iso-8859-2*) pokazuje kodiranje teksta prema standardu ISO/IEC 8859-2, poznat i kao *Latin-2* (latinična abeceda za primjenu u nekim srednjeeuropskim jezicima, uključujući hrvatski). Postoje i drugi načini (npr. stari windows-1250), ali kao moderni način kodiranja preporučuje se **utf-8**.

Unutar tijela dokumenta unosi se sadržaj, tj. tekst, hiperveze, slike i drugi statični i dinamički sadržaji. Neke osnovne HTML-oznake za oblikovanje teksta prikazane su u tablici (Tablica 1).

Tablica 1. HTML, osnovne oznake za oblikovanje teksta i primjeri

HTML kod, primjer	Izgled u pregledniku
<code>&lt;h1&gt;Naslov (prva razina)&lt;/h1&gt;</code>	<b>Naslov (prva razina)</b>
<code>&lt;h2&gt;Podnaslov (druga razina) &lt;/h2&gt;</code>	<b>Podnaslov (druga razina)</b>
<code>&lt;p&gt;Ovo je odlomak ili paragraf. Prijelom teksta je automatski, a redak se može po potrebi prelomiti bilo gdje, kao na primjer baš &lt;br&gt; tu.&lt;/p&gt;</code>	Ovo je odlomak ili paragraf. Prijelom teksta je automatski, a redak se može po potrebi prelomiti bilo gdje, kao na primjer baš tu. (*)
Tekst je ovako <code>&lt;b&gt;masno otisnut&lt;/b&gt;.</code>	Tekst je ovako <b>masno otisnut</b> .
Tekst je ovako <code>&lt;i&gt;otisnut kurzivom&lt;/i&gt;.</code>	Tekst je ovako <i>otisnut kurzivom</i> .
<code>&lt;ul&gt;     &lt;li&gt;prva natuknica&lt;/li&gt;     &lt;li&gt;druga natuknica&lt;/li&gt; &lt;/ul&gt;</code>	<ul style="list-style-type: none"> <li>• prva natuknica</li> <li>• druga natuknica</li> </ul>
<code>&lt;ol&gt;     &lt;li&gt;prva natuknica&lt;/li&gt;     &lt;li&gt;druga natuknica&lt;/li&gt; &lt;/ol&gt;</code>	<ol style="list-style-type: none"> <li>1. prva natuknica</li> <li>2. druga natuknica</li> </ol>

\* U HTML-u se tolerira se i korištenje oznake `<p>` bez zatvaranja s `</p>`.

Najvažnije kontrolne oznake su one kojima se postavljaju hiperveze. Sintaksa hiperveze uvijek koristi URI (URL) za „adresiranje“ informacijskog resursa, i neki tekst ili sliku kao vidljivu naznaku hiperveze. Tekst je često vizualno naglašen podcrtavanjem ili bojom, dok slika obično nedvosmisleno upućuje na sadržaj (na primjer, oznaka na karti, reklama, i slično). Osnovna sintaksa hiperveze je:

```
<a href="uri">hiperveza</a>
```

pri čemu je *uri* najčešće apsolutni URI na drugi resurs ili relativni URI u odnosu na trenutni


dokument. Primjeri HTML-oznaka za uvođenje hiperveza prikazani su u tablici (Tablica 2).

Tablica 2. HTML, osnovne oznake za unošenje hiperveza i drugih objekata

HTML kod, primjer	Izgled u pregledniku
<b>Hiperveza:</b> Pogledajte na stranicama FER-a, <code>&lt;a href="http://www.fer.hr"&gt;ovdje&lt;/a&gt;.</code>	Pogledajte na stranicama FER-a, <a href="http://www.fer.hr">ovdje</a> . (apsolutni URL)
Ovaj link vodi na sljedeću stranicu <code>&lt;a href="./str2.html"&gt;dalje&lt;/a&gt;.</code>	Ovaj link vodi na sljedeću stranicu <a href="#">dalje</a> . (relativni URL)
<code>&lt;a name="#vrh"&gt;...&lt;/a&gt;</code> ... Povratak na <code>&lt;a href="#vrh"&gt;vrh</code> stranice <code>&lt;/a&gt;.</code>	Oznaka imena dolazi na „odredište“ skoka preko hiperveze.  Povratak na <a href="#">vrh stranice</a> . (imenovani URL)

U dokument se mogu uvoditi i tablice, slike i razni grafički elementi, video, zvuk i drugo (na primjer, Java applet za prikaz SMIL prezentacije). Tablica 3 prikazuje HTML oznake za uvođenje tablica i slika.

Tablica 3. HTML, osnovne oznake za uvođenje slika i tablica i primjeri

HTML kod, primjer	Izgled u pregledniku						
<b>Slika (&lt;img src="url"&gt;):</b> <code>&lt;img src="./icon.gif"&gt;</code>  ili - još bolje - s alternativnim tekstom za ne-tekstualne preglednike, kao i posebne preglednike za slabovidne i slijepe osobe: <code>&lt;img src="./icon.gif" alt="[FER logo]"&gt;</code>	  [FER logo]						
<b>Tablica:</b> <code>&lt;table border="1"&gt;</code> <code>  &lt;tr&gt;</code> <code>    &lt;th&gt;Prvo zaglavlje&lt;/th&gt;</code> <code>    &lt;th&gt;Drugo zaglavlje&lt;/th&gt;</code> <code>  &lt;/tr&gt;</code> <code>  &lt;tr&gt;</code> <code>    &lt;td&gt;redak 1, stupac 1&lt;/td&gt;</code> <code>    &lt;td&gt;redak 1, stupac 2&lt;/td&gt;</code> <code>  &lt;/tr&gt;</code> <code>  &lt;tr&gt;</code> <code>    &lt;td&gt;redak 2, stupac 1&lt;/td&gt;</code> <code>    &lt;td&gt;redak 2, stupac 2&lt;/td&gt;</code> <code>  &lt;/tr&gt;</code> <code>&lt;/table&gt;</code>	<table border="1"> <thead> <tr> <th>Prvo zaglavlje</th><th>Drugo zaglavlje</th></tr> </thead> <tbody> <tr> <td>redak 1, stupac 1</td><td>redak 1, stupac 2</td></tr> <tr> <td>redak 2, stupac 1</td><td>redak 2, stupac 2</td></tr> </tbody> </table>	Prvo zaglavlje	Drugo zaglavlje	redak 1, stupac 1	redak 1, stupac 2	redak 2, stupac 1	redak 2, stupac 2
Prvo zaglavlje	Drugo zaglavlje						
redak 1, stupac 1	redak 1, stupac 2						
redak 2, stupac 1	redak 2, stupac 2						

Većina HTML oznaka može imati i atribut. Primjeri atributa su boja pozadine, boja teksta, veličina i oblik slova, poravnanje teksta, širina i visina slike, format tablice, i slično.

Na primjer, za tekst kojem bismo htjeli definirati oblik, boju i veličinu, trebali bismo uvesti stil:

```
<p style="font-family:times; font-size:120%; color:green">Ovaj odlomak
teksta obojan je zeleno. Ovaj odlomak teksta obojan je zeleno. </p>
```

.. i tako za svaki dio teksta, i u svakoj stranici gdje se želi slijediti isti izgled. Očito je da je ovo vrlo nepraktično, osobito ako se koristi samo „čisti HTML“.

Kako bi se sadržaj odvojio od formatiranja te radi jednostavnosti primjene i rukovanja izgledom stranice i atributima, u modernom dizajnu weba primjenjuje se tehnologija *Cascading Style Sheets* (CSS). Glavna prednost CSS-a je da dizajneru omogućuje definiranje stilova koji se mogu uniformno primijeniti na cijeli niz stranica koje čine web-sjedište.

Trend razdvajanja sadržaja i formata nastavlja se primjenom jezika eXtensible Markup Language (XML) za kodiranje „najnovije generacije“ HTML-a – XHTML-a.

## 2.1 Primjeri i zadaci

Primjeri i zadaci postavljeni su u vršni direktorij dokumenata web poslužitelja u emuliranoj okolini (v. poglavlje 6.1, Pokretanje mrežne okoline). **Ako nije drugačije navedeno, svi direktoriji spomenuti u ovim uputama nalaze se unutar tog direktorija.**

Izvorni kod iz primjera na slici 1 i primjeri koji ilustriraju primjenu osnovnih HTML oznaka nalaze se u sljedećim direktorijima:

- *primjer-html*, HTML dokument i primjeri HTML koda
- *primjer-css*, primjeri HTML koda, s korištenjem CSS-a

Proučite primjere. U zadacima koji slijede tražit će se da samostalno kreirate HTML dokument iz zadanog teksta.

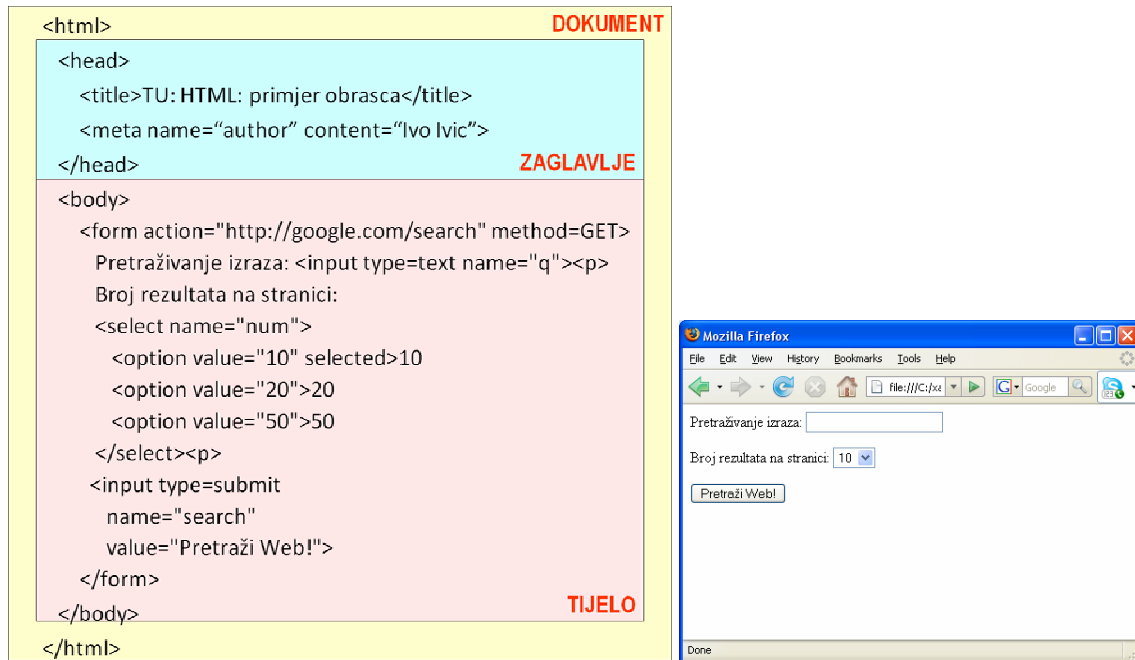
## 3 Osnove kodiranja HTML obrazaca

Obrasci su posebni HTML dokumenti, kojima je moguće prikupiti unos raznih podataka od korisnika. Obrazac se nalazi u tijelu dokumenta i može sadržavati razne elemente za unos, kao što su tekstualno polje, padajući izbornik, isključivi izbor (engl. *radio buttons*), višestruki izbor (engl. *checkbox*), skriveni unos i unos lozinke. Osnovna sintaksa obrasca je:

```
<form >
  <input>
  <input>
</form>
```

pri čemu *form* označava početak i kraj obrasca, a polje *input* služi za korisnikov unos. Oznaka *form* ima atribut *action* i *method* kojima se zadaju URI resursa i HTTP metoda pristupa.

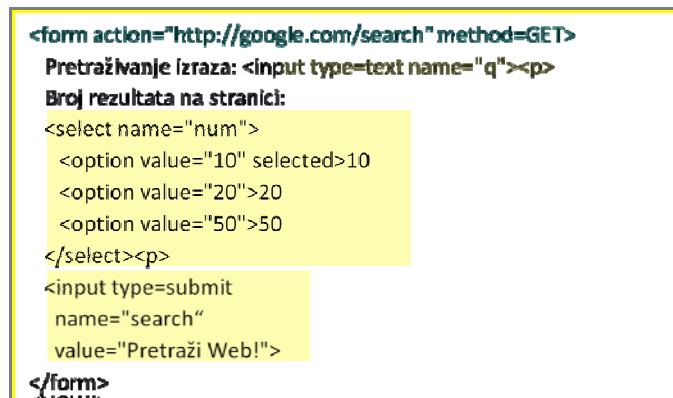
Slika 2 pokazuje primjer dokumenta koji sadrži obrazac (primjer je s predavanja), odnosno izvorni HTML kod datoteke i izgled u pregledniku.



Slika 2. Primjer HTML obrasca

Slika 3 detaljnije ilustrira dijelove obrasca u primjeru. Obrazac omogućuje korisniku unos izraza za pretraživanje, broj rezultata na stranici i slanje podataka resursu definiranom URI-jem *http://google.com/search*. Resurs u pitanju je očito tražilica *Google*. Stoga će ovaj primjer raditi *samo ako klijent ima pristup Internetu*, odnosno, ako može pristupiti tom resursu (što nije slučaj ako se radi isključivo u emuliranoj okolini!).

Kad korisnik klikne na gumb „Pretraži Web!“, sadržaj obrasca šalje se resursu definiranom u polju *action*, metodom zadanom pod *method* (ovdje, GET). Ovaj obrazac ima tri polja za unos: tekstualno polje za zadavanje izraza za pretraživanje, padajući izbornik sa zadanim brojem rezultata te gumb za slanje podataka.



Slika 3. HTML oznake u obrascu

Tablica 4 ilustrira još neke vrste unosa koje se koriste u obrascima.

Tablica 4. Primjeri polja za korisnikov unos

HTML kod, primjer	Izgled u pregledniku
<pre>&lt;form&gt; Ime i prezime: &lt;input type="text" name="ime"&gt; &lt;/form&gt;</pre>	<p>Tekstualno polje:</p> <p>Ime i prezime: <input type="text"/></p>
<pre>&lt;form&gt; &lt;input type="radio" name="spol" value="male"&gt; Muski &lt;br&gt; &lt;input type="radio" name="spol" value="female"&gt; Zenski &lt;/form&gt;</pre>	<p>Isključivi izbor:</p> <p><input type="radio"/> Muski</p> <p><input type="radio"/> Zenski</p>
<pre>&lt;form&gt; Imam automobil: &lt;input type="checkbox" name="vozilo" value="automobil"&gt; &lt;br&gt; Imam bicikl: &lt;input type="checkbox" name="vozilo" value="bicikl"&gt; &lt;/form&gt;</pre>	<p>Višestruki izbor:</p> <p>Imam automobil: <input type="checkbox"/></p> <p>Imam bicikl: <input type="checkbox"/></p>

Smisao obrazaca je da se na temelju unešenih podataka na poslužitelju može izvesti neka radnja. Logika pojedinih akcija ovisit će o primjeni. Na temelju analize zahtjeva, dizajner web sjedišta mora osmisliti i programski izvesti logiku aplikacije. Analiza uključuje cijeli ciklus: koje podatke prikupiti, što s njima napraviti, kako pokazati rezultat korisniku. Primjer skripte na strani poslužitelja bit će pokazan u sljedećem poglavlju.

### 3.1 Primjeri i zadaci

Primjer obrasca i elementi obrasca iz ovog poglavlja nalaze se u direktoriju *primjer-obrazac* unutar vršnog direktorija web poslužitelja u emuliranoj okolini (v. poglavlje 6.1, Pokretanje mrežne okoline). Proučite primjere. U zadacima koji slijede tražit će se da samostalno napravite HTML obrazac za unaprijed zadanu skriptu na strani poslužitelja.

## 4 Osnove skripti na poslužitelju

Skripte na poslužitelju moćan su alat za izradu aplikacija weba. Postoje razne poslužiteljske tehnologije, na primjer Common Gateway Interface (CGI), Java servleti, JavaServer Pages (JSP), Microsoftova tehnologija Active Server Pages (ASP), Perl, PHP i drugi. U praksi su posebno korisne *troslojne aplikacije*, koje osim web poslužitelja i logike aplikacije uključuju i bazu podataka. Kombinacijom tih triju komponenata mogu se izvesti vrlo složene aplikacije. S



obzirom da se skripte izvode na poslužitelju, neophodna je izuzetna pažnja o dizajnu skripti i utjecaju na sigurnost sustava te stalna briga o sigurnosnim mehanizmima, od mrežnih (*firewall*, aplikacijski *gateway*) do zakrpa na web poslužitelju (povrh svih uobičajenih mehanizama).

U sljedećih nekoliko primjera koristit ćemo gotove skripte. U emuliranoj okolini ove skripte smještene su u direktoriju *primjer-cgi* u vršnom direktoriju web poslužitelja.

Prvi primjer pokazuje skriptu u jeziku Perl koja ispisuje poruku „Hello world!“. Izvorni kod, s manjim modifikacijama, nalazi se u direktoriju *primjer-cgi*, u datoteci *helloworld.pl*.

```
#!/usr/bin/perl

print "Content-type: text/html\r\n\r\n";
print "<html>\n";
print "<head><title>Hello World!</title></head>\n";
print "<body>\n";
print "<h2>Hello World!</h2>\n";
print "</body>\n";
print "</html>\n";

exit (0);
```

Možemo uočiti tri ključne komponente CGI programa: 1) oznaku vrste sadržaja (*MIME type*) – ovdje, *text/html*, 2) HTML dokument, i 3) vrijednost koju program vraća (sukladno uobičajenim konvencijama, i ovdje nula obično znači ishod bez pogrešaka). U emuliranoj mrežnoj okolini, skriptu možete pokrenuti preko odgovarajućeg URI-ja. Rezultat izvođenja skripte je HTML datoteka (pogledati View/Source).

Drugi primjer je nešto složenija skripta koja obrađuje korisnikov unos i ispisuje vrijednosti.

Slika 4. Skripta *mojaperlskripta.pl*

```
#!/usr/bin/perl

use strict;
use CGI;
my $cgi = new CGI;
print
    $cgi->header() .
    $cgi->start_html( -title => 'Rezultati' ) .
    $cgi->h1('Rezultati') . "\n";
my @params = $cgi->param();
print '<TABLE border="1" cellspacing="0" cellpadding="5">' . "\n";
foreach my $parameter (@params) {
    print "<tr><th>$parameter</th><td>" . $cgi->param($parameter) .
    "</td></tr>\n";
}
print "</TABLE>\n";
print $cgi->end_html . "\n";
exit (0);
```

Obrazac kojim se zadaju podaci je pokazan na slici (Slika 5):

Slika 5. Obrazac *mojobrazac.html* kojim se predaju podaci skripti *mojaperskripta.pl*

Treći primjer u direktoriju je skripta *tutorial.pl*, koja kreira obrazac i obrađuje korisnikov unos.

#### 4.1 Primjeri i zadaci

Primjer obrasca i elementi obrasca iz ovog poglavlja nalaze se u direktoriju *primjer-cgi* unutar vršnog direktorija web poslužitelja u emuliranoj okolini (v. poglavlje 6.1, Pokretanje mrežne okoline). Proučite primjere.

## 5 Osnove skripti na klijentu

Skripte na klijentu obično se koriste za prikaz dinamičkih elemenata i interakciju s korisnikom. Skripte se mogu nalaziti i u zaglavlju i u tijelu HTML dokumenta. Razlika je u vremenu učitavanja: skripte u tijelu (`<body>`) dokumenta izvode se prilikom učitavanja stranice, dok se skripte u zaglavlju (`<head>`) izvode na poziv.

Nekoliko primjera česte primjene skripti su:

- dinamički promjenjivi elementi (npr., izbornici, reklame)

- provjera korisnikovog unosa (npr., je li korisnik unio sadržaj obaveznog polja),
- provjera unešenih podataka (npr., je li podatak odgovarajuće vrste - numerički, slovnici, formatirani broj telefona, e-mail adresa, itd.),
- brisanje svih polja u obrascu (npr., za novi unos),
- potvrda unosa (npr., iskačući prozorčić za upozorenje (npr. „jeste li sigurni“)).

Skripte u HTML definiraju se pomoći oznake `<script>`. Atribut `type` daje informaciju o skriptnom jeziku. Ova jednostavna skripta u Javascriptu ispisuje poruku „Hello world“:

```
<html>
<head></head>
<body>

<script type="text/javascript">
document.write("Hello World!")
</script>

</body>
</html>
```

Za preglednike koji ne podržavaju skripte, korisno je ubaciti oznaku `<noscript>`, u koju se može upisati poruka, na primjer:

```
<noscript>Your browser does not support JavaScript!</noscript>
```

Skripte se, kao i CSS, mogu nalaziti i odvojeno od dokumenta. Na primjer, ako bi se skripta iz primjer nalazila u datoteci *hello.js*, u HTML dokumentu u *script* oznaci stavila bi se samo referenca na skriptu u atribut `src`.

```
<script src="hello.js">
```

Primjer script elementa ugrađenog u HTML dokument se nalazi u direktoriju primjer-script, u datoteci *helloworld.html*.

Kad je riječ o skriptama, treba spomenuti da one - opravdano, ali samo donekle neizbježno - imaju etiketu „sigurnosnih rupa“, što zbog programerskih pogrešaka prilikom izvedbe skripte, što zbog skrivenih ranjivosti u preglednicima. Sudeći prema učestalosti izlaženja zakrpa za web preglednike, opasnost je stvarna. Sigurnosni model JavaScripta zasnovan je na Javi, tako da bi se skripte, teorijski, trebale izvoditi u ograničenom „pješčaniku“ (engl. *sandbox*) izolirano od ostatka operacijskog sustava. Sigurnosne politike preglednika ograničavaju ili onemogućavaju akcije kao što su pristup lokalnom disku, memorijskim lokacijama drugih program ili operacijskog sustava. Na žalost, u praksi su (naravno) poznati primjeri i načini kako zlonamjerne skripte i/ili zlonamjerni korisnici mogu napraviti itekakvu štetu.

## 6 Zadaci

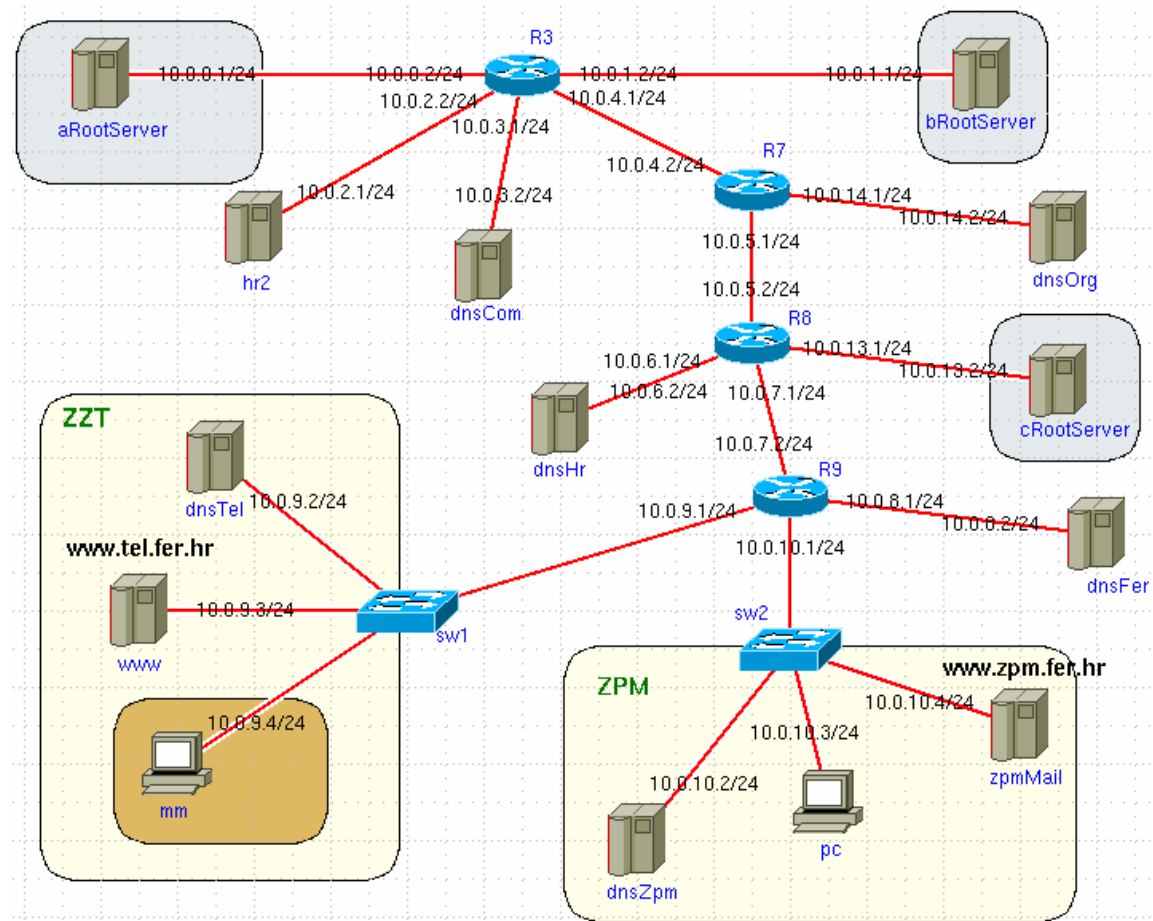
U zadacima koji slijede kreirat ćete više dokumenata, a rješenja ćete na kraju vježbe postaviti na Moodle.

## 6.1 Pokretanje mrežne okoline

Mrežnu okolinu zamijeniti će alat IMUNES. Pokrenite Vmware Player, učitajte virtualni stroj FreeBSD.vmx i nakon inicijalizacije sustava pokrenite shell, odn. *xterm*. Postavite radni direktorij na direktorij s primjerima i pokrenite IMUNES:

```
# cd /root/Examples/WEB
# imunes WEB.imn &
```

Nakon pokretanja IMUNES-a, topologija mreže pokazana je slikom (Slika 6).



Slika 6. Mrežna topologija u IMUNES-u

Pokrenite eksperiment (Experiment/Execute) i pričekajte da se eksperiment uspješno inicijalizira (pratite ispis u statusnoj traci u dnu IMUNES prozora).

Nakon toga, pokrenite DNS poslužitelje. Kako biste obrisali prethodnu konfiguraciju, unesite:

```
# rm /etc/namedb
# rm /etc/resolv.conf
```

DNS poslužitelji se pokreću pomoću shell skripte *start\_dns* (opet, pričekajte da se ovo uredno dovrši, što traje nekih desetak sekundi):

```
# ./start_dns
```

Tablica 5 daje pregled domena u virtualnoj topologiji. Konfiguracija DNS poslužitelja nalazi se u direktorijima: `aRootServer`, `bRootServer`, `cRootServer`, `dnsCom`, `dnsOrg`, `dnsHr`, `hr2`, `dnsFer`, `dnsTel` i `dnsZpm`.

Tablica 5. Popis DNS domena, poslužitelja i naziva u domeni

Domena	Poslužitelj	Nazivi u domeni
.	aRootServer, bRootServer i cRootServer	
com.	dnsCom	
org.	dnsOrg	
hr.	dnsHr	
fer.hr.	dnsFer	
tel.fer.hr.	dnsTel	www mm
zpm.fer.hr.	dnsZpm	zpmMail www = zpmMail pc

U pripremljenoj konfiguraciji web poslužitelji pokreću se „automatski“, tako da ih ne trebate posebno pokretati. (Konfiguracija se inače nalazi u „Custom startup config“ opciji u konfiguraciji hostova `www` (`www.tel.fer.hr`) i `zpmMail` (`www.zpm.fer.hr`)). Programska podrška Web poslužitelja je *lighttpd*.

Konfiguracijske datoteke za *lighttpd* su:

- `zzt.lighttpd.conf`, za poslužitelj `www.tel.fer.hr` i
- `zpm.lighttpd.conf`, za poslužitelj `www.zpm.fer.hr`.

Vršni direktorij u kojem se nalaze dokumenti poslužitelja (*server.document-root*) se nalaze u poddirektorijima od **`/root/Examples/WEB/`**:

- `www.zzt`
- `www.zpm`

S obzirom da virtualizirana okolina koristi isti datotečni sustav, direktoriji oba poslužitelja će biti istovremeno vidljivi administratoru (*root*), tako da se datoteke mogu kreirati, kopirati, modificirati, premješati, itd. iz *xterm* konzole. U stvarnosti bi svaki administrator imao pristup samo onim datotekama koje se nalaze u datotečnom sustavu lokalnog poslužitelja.

S „mrežne strane“ se web poslužiteljima pristupa preko URI-ja koji vode svaki na svoju početnu datoteku *index.html* u pripadajućem vršnom direktoriju, dakle vrijedi:

- URI **`http://www.tel.fer.hr`** vodi na **`/root/Examples/WEB/www.zzt/index.html`**
- URI **`http://www.zpm.fer.hr`** vodi na **`/root/Examples/WEB/www.zpm/index.html`**

Provjera rada web poslužitelja može se obaviti grafičkim web-preglednikom Opera, ili pomoću tekstualnog preglednika *elinks*. Ako se koristi grafički web-preglednik Opera, treba odabrati «Right-click» izbornik na čvoru PC (ili nekom drugom), odabrati «Opera Browser» i,

nakon što se preglednik pokrene, upisati željeni URL, na primjer, <http://www.tel.fer.hr/>.

**VAŽNO:** U svakom trenutku može biti pokrenuta samo jedna instanca preglednika Opera.

Ako se koristi tekstualni web preglednik `elinks`, treba otvoriti izbornik (desni-klik mišem) na čvoru PC, odabrati *Shell window/Bash* i u prozoru terminala pozvati naredbu:

```
pc# elinks www.tel.fer.hr
```

Na poslužitelju `www.tel.fer.hr` postavljena je početna stranica (uobičajeno nazvana *index.html*). Na toj stranici nalaze se definiran je link na «formular», tj. primjer za CGI skriptu, te link na stranicu drugog poslužitelja, [www.zpm.fer.hr](http://www.zpm.fer.hr).

Prije nastavka rada, provjerite da je inicijalizacija sustava uspješno provedena. Na računalu `pc.zpm.fer.hr` pokrenite preglednik Opera. Trebali biste dobiti početnu stranicu (Slika 7).



Slika 7. Web preglednik *Opera* a) početna stranica i b) naslovnica `www.tel.fer.hr`

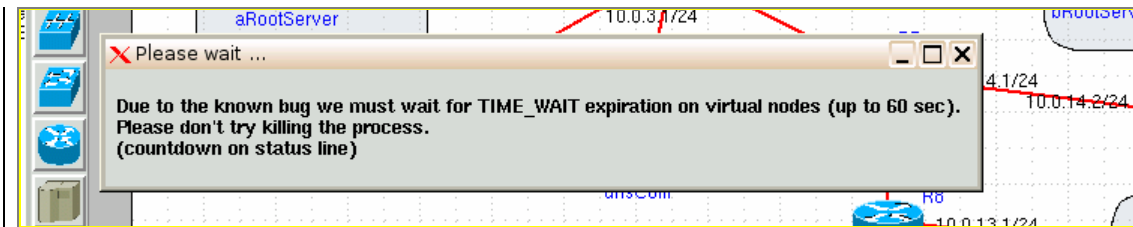
U adresno polje upišite URL <http://www.tel.fer.hr>. Kad se učita početna stranica, kliknite na link Link na ZPM. Ako se i ta stranica dobro učita, to znači da ste uspješno uspostavili mrežno okruženje.

Sada slijede zadaci.

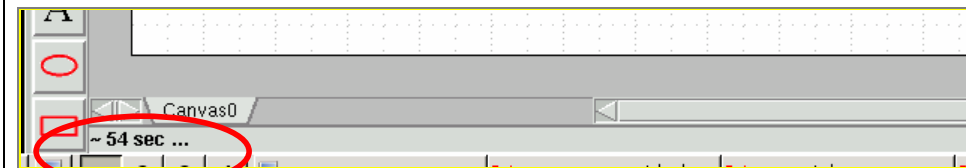
#### VAŽNA NAPOMENA:

IMUNES se ne može uredno zatvoriti za vrijeme izvođenja eksperimenta - **pokušaj da se to nasilno napravi, na primjer zatvaranjem prozora, izaziva *reboot***. Uredno zatvaranje eksperimenta radi se pomoću opcije *Experiment/Terminate* u IMUNES-u i traje 60 sekundi (istek TCP vremenske kontrole `TIME_WAIT`).

Prilikom zaustavljanja eksperimenta pojaviti će se prozor s upozorenjem:



Kada se to dogodi, ne radite ništa; samo pričekajte istek vremenske kontrole. Maksimalno preostalo vrijeme može se vidjeti u statusnoj traci:



## 6.2 Kreiranje statičkog HTML dokumenta

U prvom dijelu vježbi kreirat ćete sadržaj na virtualnom računalu na kojem je Web poslužitelj ([www.tel.fer.hr](http://www.tel.fer.hr)). Pokrenite *shell* na tom računalu.

Radni direktorij unutar kojeg ćete pohranjivati dokumente za *X*-ti zadatak nazvan je *zadatakX*. Pozicioniranje u određeni direktorij radi se pomoću naredbe *cd* (*change directory*).

**Zadatak 1.** Kreirajte HTML dokument na WWW poslužitelju [www.tel.fer.hr](http://www.tel.fer.hr).

Postavite radni direktorij na **`/root/Examples/WEB/www.zzt/zadatak1`**.

```
% cd /root/Examples/WEB/www.zzt/zadatak1
```

Zadane su datoteke koje ćete koristiti za izradu HTML dokumenta:

- *suma\_striborova\_lektira.txt*, koja sadrži najvažnije informacije o priči „Šuma Striborova“ iz zbirke „Priče iz davnine“ spisateljice Ivane Brlić-Mažuranić,
- *osnovni.html*, početna struktura HTML dokumenta s utf-8 kodiranjem (kao predložak),
- *suma\_striborova.txt*, koja sadrži tekst priče „Šuma Striborova“,
- *suma\_striborova.html*, koja na prethodni dokument dodaje osnovne HTML oznake (i ništa više),
- *ibm.jpg*, koja sadrži sliku spisateljice u JPEG formatu.

Vaš zadatak je da tih dokumenata napravite HTML stranice, isključivo pomoću programa za uređivanje teksta. Kao program za uređivanje teksta možete koristiti NEedit ili jednostavni editor unutar opcije *View/Source* (Ctrl+F3) u web pregledniku Opera (prednost ovog drugog je vidljiv prikaz slova ć, č, đ, š, ž).

Organizacija HTML stranica zamišljena je kao osnovni dokument s podacima u datoteci

*suma\_striborova\_lektira.txt*, kojeg treba povezati sa stranicom koja sadrži tekst priče. Tekst priče *suma\_striborova.txt* treba biti podijeljen u nekoliko HTML dokumenta, međusobno povezanih hipervezama za navigaciju.

Slijedite ove smjernice za postupak izrade stanica:

- (1) Unutar direktorija *zadatak1*, kopirajte datoteku *osnovni.html* u (novu) datoteku *index.html*.  
(Oprez - nemojte zamijeniti taj *index.html* s istoimenim dokumentom u vršnom direktoriju!)
- (2) Otvorite *index.html* u pomoću programa za uređivanje teksta. Unutar zaglavlja definirajte naslov (<title>) i ime autora HTML dokumenta (svoje ime) u meta oznaci *author*.
- (3) Kopirajte tekst iz datoteke *suma\_striborova\_lektira.txt* u tijelo dokumenta.
- (4) Uredite naslov, podnaslov i dijelove teksta unošenjem HTML oznaka.

Ako ste koristili NEdit, sada pokrenite Web klijent *Opera* na računalu **www.tel.fer.hr**.

- (5) Učitajte napravljeni dokument u preglednik i provjerite njegov izgled i funkcionalnost.
- (6) Ispravite uočene pogreške.
- (7) Modificirajte dokument tako da koristi CSS stil *xampp.css*.

Nakon što je *index.html* s osnovnim podacima o autorici i priči uspješno završen, uredite i ostatak, tj. same stranice priče.

- (8) Razdvojite sadržaj u nekoliko „poglavlja“, odnosno datoteka (minimalno 3). Primijenite CSS stil *xampp.css*.
- (9) Za svako poglavlje podesite <title> oznaku i unesite naslov ili podnaslov za taj dio priče.
- (10) Osmislite i unesite hiperveze koje će služiti za navigaciju (početak-kraj, naprijed-natrag) između pojedinih poglavlja. Prilikom unošenja hiperveza koristite isključivo lokalne URI-je.
- (11) Učitajte napravljene dokumente u preglednik i provjerite njihov izgled i funkcionalnost. Ispravite uočene pogreške.
- (12) Kad ste potvrdili da je dokument ispravan, zatvorite klijenta *Opera*.

Pokrenite Web klijent *Opera* na računalu **pc.zpm.fer.hr**. Pazite da u datom trenutku unutar IMUNES-a bude pokrenuta samo jedna instanca preglednika Opera (ako postoje ranije otvorene instance, zatvorite ih prije nastavka).

Napomena: Ako se dokument pokazao ispravnim u lokalnom radu, ali se preko mreže ne učitava kako treba, provjerite ovlasti čitanja, pisanja i izvođenja za pristup dokumentima



(pretpostavljene vrijednosti su: 755 za direktorije i izvršne datoteke i 644 za obične datoteke). Ovlasti se mijenjaju pomoću naredbe *chmod* (pogledati *%man chmod* za upute).

Odgovorite na pitanja:

- a) Koji je URI za pristup vršnom direktoriju *www* poslužitelja *www.tel.fer.hr*?
- b) Koji je URI za pristup dokumentu *index.html* iz ovog zadatka?

Sada zatvorite eksperiment (Experiment/Terminate). NE PREKIDAJTE uredno zatvaranje (pričekajte 60 s). Zatvorite IMUNES kako biste obrisali sve priručne spremnike za DNS.

**Zadatak 2.** Proučite ponašanje protokola TCP i HTTP u mreži prilikom uspješne komunikacije klijenta i poslužitelja.

Ponovno pokrenite mrežnu okolinu kako je opisano u poglavlju 6.1. Pokrenite Web klijent *Opera* na računalu **pc.zpm.fer.hr**. Pazite da u datom trenutku unutar IMUNES-a bude pokrenuta samo jedna instanca preglednika *Opera* (ako postoje ranije otvorene instance, zatvorite ih prije nastavka).

- (1) Pokrenite Wireshark/ethereal na istom računalu i pokrenite snimanje paketa.
- (2) U pregledniku *Opera* učitajte dokument iz prethodnog zadatka.
- (3) Nakon što je stranica uspješno učitana, zaustavite snimanje paketa.
- (4) Filtrirajte samo pakete vezane za ovu komunikaciju (TCP, HTTP i DNS).
- (5) Pohranite samo filtrirane pakete (File/Save as...) u datoteku *zad2.pcap* u direktoriju *zadatak2*.

Nemojte zatvarati preglednik, samo ga privremeno minimizirajte, jer će nam trebati u sljedećem zadatku. Na temelju analize snimljenih paketa, odgovorite na pitanja:

- a) Ispišite IP adrese i brojeve TCP vrata klijenta i poslužitelja.
- b) Koliko se TCP konekcija uspostavilo prilikom dohvaćanja stranice?
- c) Kada i zašto klijent komunicira s DNS poslužiteljem?
- d) Koja verzija protokola HTTP (1.0 ili 1.1) se koristi u komunikaciji?
- e) Koja HTTP metoda se koristi za pristup resursu, GET ili POST?
- f) Skicirajte razmjenu HTTP poruka prilikom učitavanja stranice.
- g) U kojem trenutku se učitava CSS datoteka?
- h) Skicirajte strukturu i ispišite najvažnija polja *zaglavlja* HTTP *zahtjeva*.
- i) Skicirajte strukturu i ispišite najvažnija polja *zaglavlja* HTTP *odgovora*.

j) Što je sadržano u *tijelu* HTTP *zahtjeva*, a što u *tijelu* HTTP *odgovora*?

**Zadatak 3.** Proučite statusne kodove HTTP-a u slučaju neuspješne komunikacije.

Ako ste sve dobro napravili u prethodnom zadatku, pokušajte „pokvariti“ stvar kako biste demonstrirali neuspješne ishode HTTP zahtjeva za dohvaćanjem resursa. Eksperiment ćemo napraviti na kopiji direktorija *zadatak1* iz prvog zadatka, koji ćemo prigodno nazvati *zadatak3*. Osnovna sintaksa naredbe za kopiranje (*cp*) je:

```
% cp <file1> <file2>          za pojedinačne datoteke
% cp -r <dir1> <dir2>         za rekurzivno kopiranje direktorija
```

Postavite radni direktorij na */root/Examples/WEB/www.zzt* i kopirajte direktorij *zadatak1* u direktorij *zadatak3*. Postavite radni direktorij na *zadatak3*.

```
% cd /root/Examples/WEB/www.zzt
% cp ./zadatak1/* ./zadatak3
% cd zadatak3
```

Napomena: Ako niste vješti u korištenju Unixa, naredba “man” (od *manual*) daje detaljne upute o sintaksi pojedinih naredbi (% man cp).

Postavite *zadatak3* kao radni direktorij.

- (1) Promijenite ovlasti pristupa datoteci *ibm.jpg* u 600 (rw-----).  

```
% chmod 600 ibm.jpg
```
- (2) U pregledniku *Opera*, obrišite priručni spremnik preglednika (engl. browser cache). Ta opcija se nalazi pod *Tools/Delete private data*.
- (3) Pokrenite Wireshark/ethereal, učitajte URI početne stranice *index.html* (iz radnog direktorija) u preglednik *Opera* i nakon učitavanja stranice, zaustavite Wireshark (koraci 1-3 iz prethodnog zadatka).
- (4) Filtrirajte samo HTTP zahtjeve i odgovore vezane za ovu komunikaciju.
- (5) Pohranite samo filtrirane pakete (*File/Save as...*) u datoteku *zad3a.pcap* u direktoriju *zadatak3*.

Odgovorite na pitanja:

- a) Što opažate s motrišta korisnika?
- b) Je li se što se promijenilo u HTTP porukama? Objasnite.
- c) Pokrenite program za uređivanje teksta NEdit i pomoću njega otvorite HTTP *access\_log* i *error\_log*.
  - access log nalazi se u */var/log/lighttpd.access.log*
  - error log nalazi se u */var/log/lighttpd.access.log*

Koji podaci su ostali zabilježeni? Komentirajte.

Nastavite sa sljedećim naredbama:

- (6) Vratite ovlasti pristupa datoteci *ibm.jpg* u 644 (rw-r--r--).  

```
% chmod 644 ibm.jpg
```
- (7) Preimenujte datoteku *index.html* u *index1.html*.  

```
% mv index.html index1.html
```
- (8) U pregledniku *Opera*, obrišite priručni spremnik preglednika (engl. browser cache). Ta opcija se nalazi pod *Tools/Delete private data*.
- (9) Pokrenite Wireshark/ethereal, učitajte URI početne stranice *index.html* u preglednik i nakon učitavanja stranice, zaustavite Wireshark (koraci 1-3 iz zadatka 2).
- (10) Filtrirajte samo HTTP zahtjeve i odgovore vezane za ovu komunikaciju.
- (11) Pohranite samo filtrirane pakete (*File/Save as...*) u datoteku *zad3b.pcap* u direktoriju *zadatak3*.

Odgovorite na pitanja:

- d) Što opažate s motrišta korisnika?
- e) Je li se što se promijenilo u HTTP porukama? Objasnite.
- f) Otvorite HTTP *acces\_log* i *error\_log*. Koja je razlika ovog i prethodnog slučaja?

### 6.3 Kreiranje HTML obrasca

**Zadatak 4.** Osmislite i napraviti HTML obrazac kojim će se unositi podaci o literatnim djelima.

U direktoriju *zadatak4* nalaze se kopije ranije korištenih datoteka: CSS datoteka *xampp.css*, obrazac *mojobrazac.html* i skripta *mojaperlskripta.pl* iz direktorija *primjer-cgi*. Obrazac *mojobrazac.html* možete koristiti kao predložak za izradu zadanog obrasca.

- (1) Promijenite obrazac tako da koristi CSS stil *xampp.css*.
- (2) Novi obrazac treba imati sljedeća polja:  
 Ime i prezime autora – tekstualno polje  
 Naslov djela – tekstualno polje  
 Razdoblje – padajući izbornik (18. st., 19. st., 20. st., 21. st.)  
 Vrsta – isključivi izbor (priča, pjesma)  
 Publika – višestruki izbor („za djecu“, „za mladež“, „za odrasle“)

Gumb za čišćenje obrasca

Gumb za slanje podataka

(3) Podaci se šalje skripti *mojaperlskripta.pl* pomoću HTTP GET metode.

(4) Provjerite rad obrasca i ispravite uočene pogreške.

Kad ste se uvjerali da obrazac ispravno radi, snimite komunikaciju klijenta i poslužitelja pomoću Wiresharka, isto kao u zadacima 2 i 3.

(5) Demonstrirajte rad obrasca unošenjem podataka o priči „Šuma Striborova“.

(6) Filtrirajte HTTP poruke i pohranite snimljene pakete u *zad4.pcap* u direktoriju *zadatak4*.

Odgovorite na pitanja:

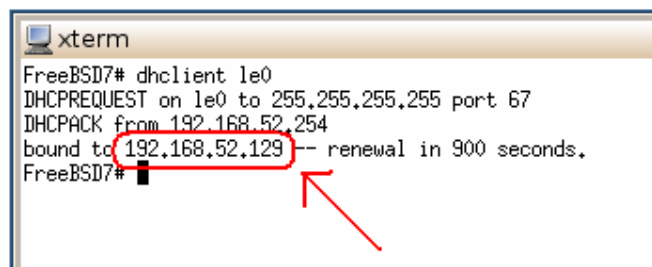
- Kako izgleda URI koji se šalje poslužitelju?
- Kako primjena CSS-a utječe na komunikaciju s poslužiteljem?
- Što bi se s motrišta korisnika, a što s motrišta poslužitelja dogodilo da zamijenimo HTTP metodu GET metodom POST?

## 7 Prebacivanje rješenja na Moodle

Da biste prebaciti rješenje na Moodle, trebate prebaciti datoteke iz virtualnog datotečnog sustava u onaj od operacijskog sustava (MS Windows). Postupak je sljedeći:

- Unutar virtualizirane okoline, u konzoli (*xterm*) pokrenite DHCP klijenta kako biste dobili IP adresu za sučelje *le0*:

```
# dhclient le0
```



Zapišite IP adresu tog sučelja.

- Unutar MS Windows okoline, pokrenite neki program pomoću kojeg možete pristupiti datotečnom sustavu virtualnog računala - *Secure Copy* (SCP), FTP ili SSH klijenta (npr., Putty).

Ako na računalu nije instaliran ni jedan od navedenih alata, možete ga instalirati u korisničkom prostoru na svom računalu.

WinSCP <http://winscp.net/eng/download.php>

SSH <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

(3) Pomoću odabranog programa spojite se na poslužitelj na IP adresi virtualnog sučelja (korisničko ime/lozinka su *imunes/imunes*) i prebacite dokumente koje ste sami izradili na lokalni disk. Podsjetnika radi, to su sljedeće datoteke, po direktorijima:

- zadatak1: *index.html* i sve HTML datoteke koji čine dijelove priče,
- zadatak2: datoteku *zad2.pcap*,
- zadatak3: datoteke *zad3a.pcap* i *zad3b.pcap*,
- zadatak4: *mojobrazac.html* i *zad4.pcap*.

Datoteke možete prebaciti pojedinačno ili kao arhivu (za arhiviranje možete koristiti naredbe *tar* ili *zip*).

(4) Napravite upload na *Moodle* na uobičajeni način.

## 8 Korisni izvori

- [1] World Wide Web Consortium [On-line: <http://w3c.org>]  
Referentni izvor za web, od početka do danas. Sve važeće specifikacije HTML-a i drugih tehnologija weba.
- [2] W3 Schools, Full Web Building Tutorials [On-line: <http://www.w3schools.com/>]  
Ovdje možete naći tutoriale za većinu tehnologija weba, uključujući HTML i Javascript.
- [3] W3C Markup Validation Service [On-line: <http://validator.w3.org/>]  
Link koji nudi besplatnu provjeru ispravnosti koda (HTML, CSS i drugo).
- [4] Javascript Security [On-line: <http://www.devarticles.com/c/a/JavaScript/JavaScript-Security/>]  
Članak o sigurnosnim rizicima i sigurnosnom modelu Javascripta.
- [5] Perl 5.10.0 documentation – CGI [On-line: <http://perldoc.perl.org/CGI.html>]