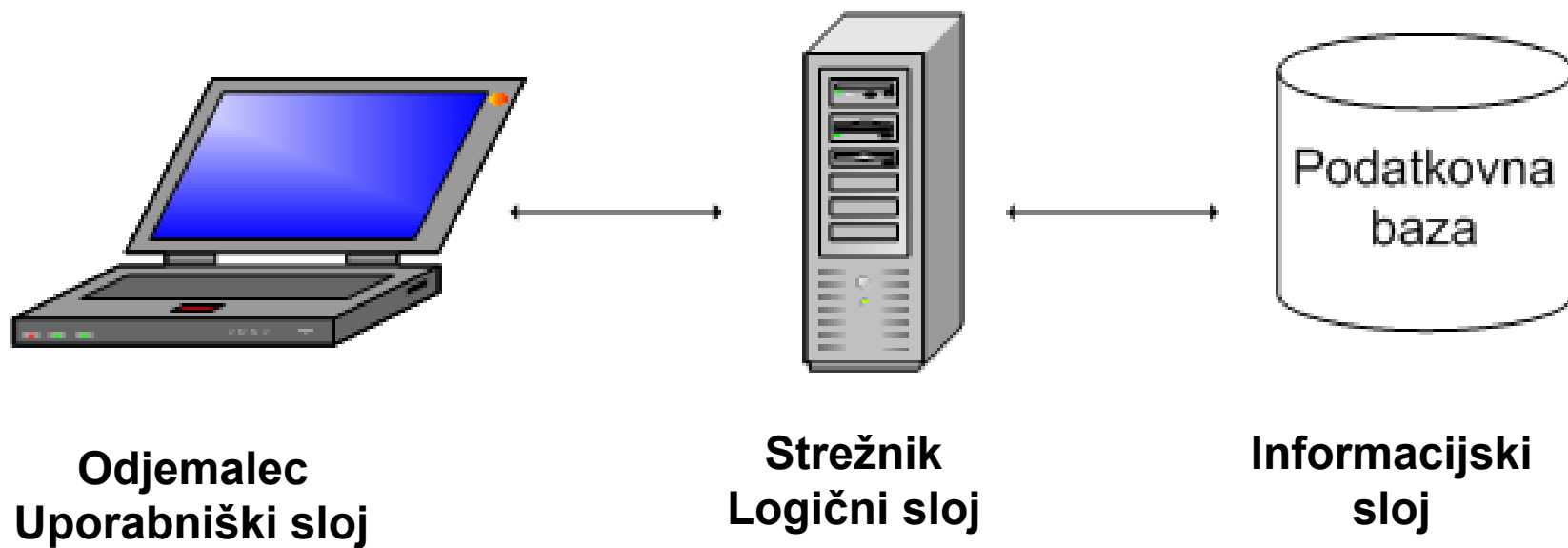


# Aktivne spletne strani – programska koda na strežniku

# Trislojna arhitektura



# Strežnik - odjemalec



**GET / HTTP/1.1**

Host: www.siol.net

User-Agent: Mozilla/5.0....

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

Accept-Language: en-us,en;...

**HTTP/1.1 200 OK**

Server: Microsoft-IIS/5.0

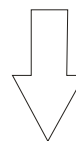
Date: Thu, 18 Sep 2008 15:17:20 GMT

X-Powered-By: ASP.NET

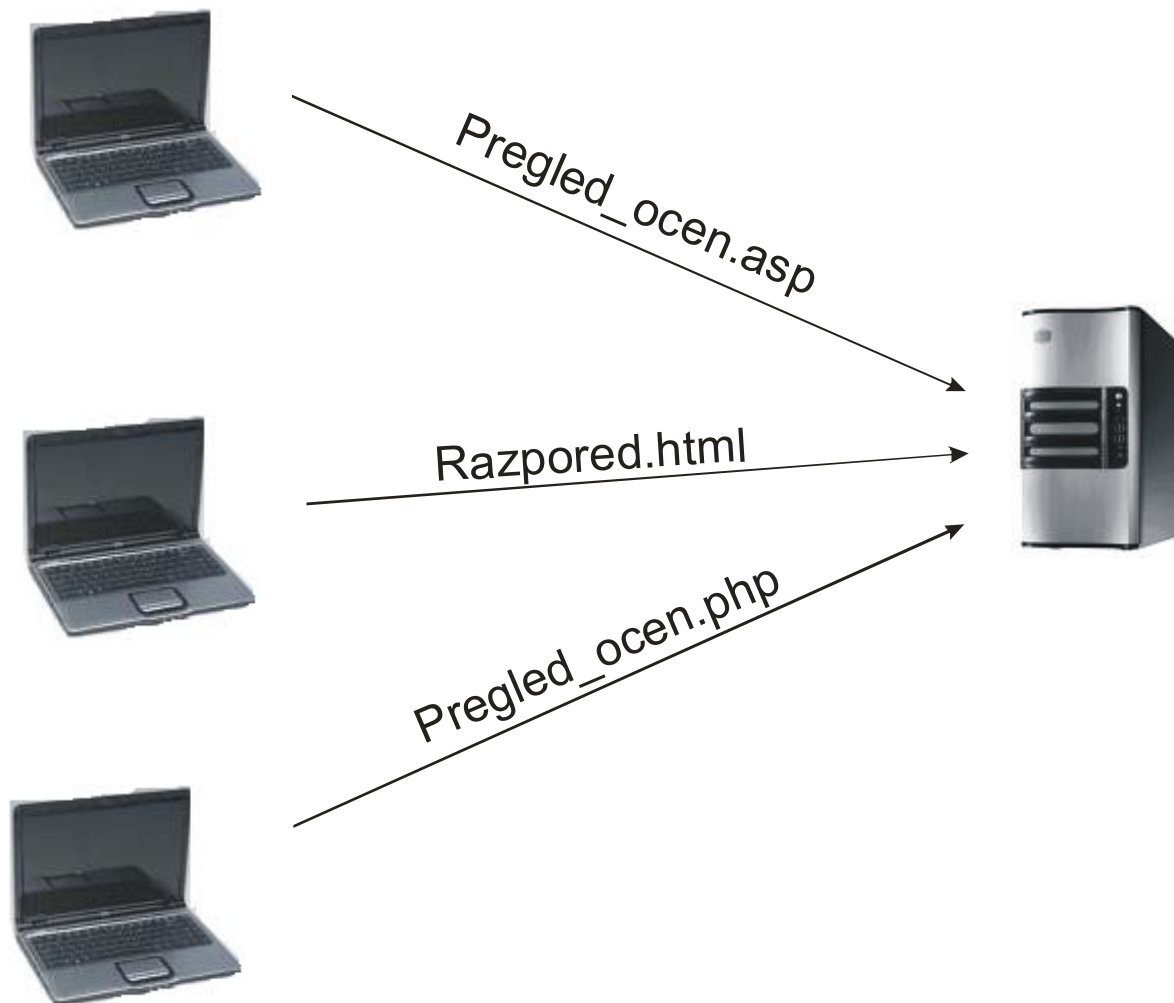
Content-Length: 4389

Content-Type: text/html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml" xml:lang="en" lang="sl">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="sl">
<head>
  <title id="head_title">SiOL.net</title>
  <meta http-equiv="content-type" content="text/html; charset="
  <link rel="stylesheet" type="text/css" href="http://itm.siol
  <link rel="stylesheet" type="text/css" href="http://itm.siol
  <script type="text/javascript" src="http://itm.siol.net/js/fi
  <link rel="alternate" type="application/rss+xml" title="RSS .
</head>
<body>
  <script src="http://utm.siol.net/u.aspx?page=PORTAL" type="t
  <div class="home" id="header">
    <div class="holder">
      <ul class="nav_corpo">
        <li class="sel"><a title="Dor
        <ul class="nav_tools">
          ...
        </ul>
      </li>
    </div>
  </div>
</body>
</html>
```



# Strežnik - odjemalec



- Programska koda na strežniku
  - Se izvede na strežniku pred pošiljanjem odgovora – strani odjemalcu
  - Odjemalec vidi le rezultat izvajanje programske kode
  - Različni programski jeziki
    - Prevedeni (C, C++, Delphi) ali skriptni (ASP, PHP, JSP, Perl, itd.)
    - Podpora s strani strežnika

- Tipične naloge programske kode na strežniku
  - Sprotna generacija vsebine spletne strani
    - Števci obiskov
    - Datum, ura in drugi “živi” podatki
  - Prilagoditev vsebine glede na zahteve ali zmožnosti odjemalca
  - Vpogledi v podatkovno bazo in prikaz rezultatov na spletni strani


- Programska koda na odjemalcu
  - Se izvede v brskalniku na odjemalcu
  - Je sestavni del HTML dokumenta in je zato vidna vsem odjemalcem
    - Ponavadi je vključena v glavi HTML dokumenta, lahko kot ločena datoteka
  - Različni programski jeziki
    - HTML, CSS
    - Javascript
    - Flash datoteke
    - ActiveX kontrole
    - Java Applet

- Tipične naloge programske kode na odjemalcu
  - Preverjanje podatkov v HTML obrazcih
  - Spreminjanje oblike strani glede na akcije uporabnika
  - Obsežne in grafično zahtevne aplikacije, ki ne potrebujejo podatkov iz strežnika



# Microsoft spletni strežnik - IIS

Tools	Third Party	Visual InterDev	Visual C++	Visual Studio.NET, Web Matrix	Visual Studio 2005, Web Developer Express
Languages	C++, Perl	VBScript JScript	C++	VBScript, JScript, VB.NET, C#	VBScript, JScript, VB.NET, C#
API/Framework		ASP	ATL Server	ASP.NET 1.0	ASP.NET 2.0
Web Server	CGI	ISAPI			
	IIS				

- Na voljo za operacijske sisteme UNIX, Linux, Mac OS in Windows
- Programski paket XAMPP  XAMPP
- Osnovne značilnosti
  - Spletni dokumenti (.../htdocs/)
  - Konfiguracija (.../conf/httpd.conf)
    - “Root” direktorij
    - “Alias”-i
    - Pravice

- ❑ Omogočajo programiranje obnašanja nekega strežnika
- ❑ Prepletanje skriptnega jezika in HTML kode
- ❑ Skripta se izvede preden se HTML dokument pošlje odjemalcu
- ❑ Glavne naloge skript
  - Dinamično spreminjanje in urejanje vsebine spletnih strani
  - Programiranje odziva na vnesene podatke uporabnika
  - Dostop do podatkovnih baz in prikaz rezultata v brskalniku na odjemalcu
  - Omogočanje varnosti, saj programska koda ni vidna odjemalcu

- ASP (Active Server Pages)
  - Programski jeziki za pisanje ASP skript
    - VBScript (poenostavljena različica Visual Basica)
    - JScript (Microsoftova različica JavaScripta)
    - Perl
  - Spletni strežniki, ki podpirajo ASP
    - Microsoft IIS
    - Sun Java System ASP
    - Personal Web Server (PWS)

- PHP (Hypertext Preprocessor)
  - Odprtokodni (open source) skriptni jezik
  - Posebna PHP sintaksa, ki je podobna Perl-u in C-ju
  - Strežniki, ki podpirajo PHP skripte
    - Apache
    - IIS 7.0 ali IIS 6.0 (z dodatnim FastCGI modulom)
  - Zmogljivejši od ASP jezika

- JSP (Java Server Pages)
  - Temelji na programskem jeziku Java
  - Strežniki, ki podpirajo JSP skripte
    - IBM WebSphere
    - Java System Application Server
    - Apache Tomcat
    - IIS (s pomočjo Tomcat-a)
  - Zelo zmogljivo orodje v kombinaciji z Java servleti in J2EE tehnologijo

# PHP: Hypertext Preprocessor

- Rasmus Lerdorf (1994)
  - Sledenje uporabnikov njegove spletne strani
  - Napisan v programskem jeziku C
- PHP 1 .. PHP 5.x
- PHP razčlenjevalnik
  - Del inštalacije Apache strežnik
  - Ločena inštalacija
- Odprtokodna tehnologija
- Enostavno povezovanje in delo z bazami



- PHP datoteka
  - HTML datoteka, kateri so dodani segmenti PHP kode
  - Tipična končnica .php (.php3 ali .phtml)
- Razlika med PHP in (X)HTML datoteko na strežniku
  - Ko brskalnik zahteva HTML datoteko, mu jo strežnik direktno posreduje
  - Ko brskalnik zahteva PHP datoteko, se ta posreduje t.i. PHP interpreterju (engine)
  - PHP interpreter obdela kodo vrstico po vrstico in izvede vse skripte
  - Odjemalcu se posreduje navadna HTML datoteka

## □ PHP sintaksa

- PHP datoteka je zelo podobna HTML datoteki, le da vsebuje php skripte
- Posamezne skripte so označene z

`<?php`

`...`

`?>`

- Obvezno zaključevanje ukazov s `;`
- Komentariji: `//` ali `/* ... */`
- Izpis besedila iz PHP: **echo** in **print**

Primer:

`echo "Izpis iz kode";`

## □ Primer enostavne PHP skripte

```
<html>
```

```
<body>
```

```
<?php
```

```
echo "Lep pozdrav"
```

```
?>
```

```
</body>
```

```
</ html >
```

## □ Spremenljivke v PHP

- Ime spremenljivke se začne z znakom \$
- Ime lahko vsebuje vse alfanumerične znake in “\_”

Primer:

```
<?php
```

```
$ime = “Janez”;
```

```
$starost = 50;
```

```
?>
```

- “loosley typed”
  - Določanje tipa ni potrebno

## □ Konstante      `define ( “starost”, 33);`

- Tipi spremenljivk
  - int, integer
  - float, double, real
  - string
  - bool, boolean
  - array
  - object
  - NULL
- Inicializacija spremenljivk
  - **undef**

## □ Določanje tipa

`gettype( $ime_spremenljivke )`

`settype( $ime_spremenljivke, "integer" )`

## □ Pretvarjanje tipa – “casting”

`(double) $ime_spremenljivke`

`(string) $ime_spremenljivke`

Ne spremeni vrednosti spremenljivke!!

Primeri:

```
$visina = "183 cm";  
settype($visina, "integer");
```

```
$visina = "183 cm";  
print( (integer) $visina);
```

## □ Aritmetični operatorji

- +, -, \*, /

`y = x*5;`

- % (po modulu), ++, --

`y=10%8;      y++;`

## □ Operatorji za prirejanje

- =, +=, -=, \*=, /=, .=, %=

`y*=x; (y=y*x;)`

## □ Operatorji za primerjanje

- ==, !=, <>, >, <, >=, <=

`2>3   3!=4   x==2`

## □ Logični operatorji

- &&, ||, !

`$x=1;`

`$y=8;`

`($x<3 && $y>5)`



- Tekstovna spremenljivka – string
  - V spremenljivki ali direktno v navednicah “...” (‘..’)
  - Združevanje z operatorjem .

```
$ime = “Branka”;      $priimek = “Novak”;
```

```
echo $ime . “ “ . $priimek . “ je v službi”;
```

- Osnovne funkcije

- strlen(\$ime)

```
strlen(“Kaj delamo tu?”); (14)
```

- strpos(\$ime, \$iskani\_string)

```
$stavek = “Janez je priden”;  
strpos($stavek, “je”); (6)
```



- Ostale funkcije za delo s string-i
  - strcmp (\$beseda1, \$beseda2);                    (-1, 0, 1)
    - strncmp(\$beseda1, \$beseda2,n);
  - strstr (\$beseda, \$iskano);                    (vrne preostanek stringa)
    - strstr (\$beseda, \$iskano);
  - trim (\$beseda, *seznam*);                    (odstrani presledke na robovih)
  - str\_shuffle (\$beseda);
  - chr(80) in ord("b");

[http://www.w3schools.com/php/php\\_ref\\_string.asp](http://www.w3schools.com/php/php_ref_string.asp)



- **Niz (array)**
  - Spremenljivka, ki hrani več vrednosti
  - Trije tipi nizov
    - Numerični (numeric)
    - Asociativni (associative)
    - Večdimenzionalni (multidimensional)

## □ Pogojni stavki

- If, if ... else, if ... elseif ... else

if (\$vrednost == 12) echo "Rezultat je 12";

If (*pogoj*)

... ;

else

... ;

If (*pogoj*)

... ;

elseif (*pogoj*)

... ;

else

... ;

## □ Pogojni stavki

### ■ switch

```
switch (pogoj);  
{  
  case rez1:  
    ... ;  
    break;  
  case rez2:  
    ... ;  
    break;  
  default:  
    ... ;  
}
```

- Programske zanke
  - for in foreach

```
for (zac; pogoj; inkrement)  
{  
    ... ;  
}
```

```
for ($i=0; $i<100; $i++;)  
{  
    print("X" . $i);  
}
```

```
foreach ($niz as $vrednost)  
{  
    ... ;  
}
```

```
$imena=array("Joc","Bor","Miha");  
foreach ($imena as $nekdo)  
{  
    print($nekdo);  
}
```

## □ Programske zanke

### ■ while

```
while (pogoj)  
{  
    ... ;  
}
```

```
while($i<=10)  
{  
    echo("Cifra:" . $i);  
    i++;  
}
```

```
do  
{  
    ... ;  
}  
while (pogoj)
```

```
do  
{  
    i++;  
    print("Cifra:" . $i);  
} while($i<=10)
```

## □ Funkcije

```
function ImeFunkcije()  
{  
    ... ;  
}
```

```
function ImeFunkcije($p1, $p2)  
{  
    ... ;  
}
```

```
<?php  
function seštej($x, $y)  
{  
    $vsota = $x+$y;  
    return $vsota;  
}
```

```
echo "12 + 23 = " . seštej(12,23);  
?>
```

- Globalne in lokalne spremenljivke / SCOPE
  - Spremenljivke, ki so definirane v funkcijah so lokalne
  - Ostale spremenljivke so globalne
  - \$GLOBALS: dostop do globalnih spremenljivk

```
<?php
    function test() {
        $ime = "Lokalno";
        echo "Iz globalne: " . $GLOBALS["ime"] . "< br />";
        echo "Iz lokalne: " . $ime;
    }

    $ime= "Globalno";
    test();
?>
```





- Obrazci
- Pošiljanje podatkov na strežnik
  - Metodi
    - GET: spremenljivke in vrednosti se prenesejo kot del URL naslova
    - POST: spremenljivke in vrednosti se prenesejo v zaglavju in niso vidne uporabniku

```
<form action="registracija.php" method="post">
```

```
    Uporabniško ime: <input type="text" name="username" />
```

```
    Geslo: <input type="password" name="password" />
```

```
    <input type="submit" value="Prijavi se"/>
```

```
</form>
```

- Dostop do podatkov iz obrazca v PHP skripti
  - Globalne spremenljivke / asociativni nizi
    - \$\_GET
    - \$\_POST
    - \$\_REQUEST
  - extract(...);
    - Ustvari posamezne spremenljivke in vrednosti na osnovi vrednosti iz asociativnega niza

**Validacija podatkov!!**

```
<?php
```

```
echo "Uporabniško ime je: " . $_POST["username"];
```

```
echo "Geslo je: " . $_POST["password"];
```

```
...
```

```
?>
```

## □ Piškotki / Cookies

- Koščki informacij, ki jih strežnik shrani na odjemalčevem računalniku
  - Se avtomatsko pošljejo na strežnik ob ponovnem obisku
- Tekstovne datoteke brez zaščite
- Datum veljavnosti
- Strežnik ima dostop samo do svojih piškotkov

## □ Piškotki

### ■ Ustvarjanje piškotka

```
<?php  
    setcookie("uporabnik", "Janez Novak", time()+3600);  
?>
```

Čas v sekundah



### ■ Branje piškotka

```
<?php  
    // Izpit piškotka  
    echo $_COOKIE["uporabnik"];  
  
    // Izpit vseh piškotkov  
    print_r($_COOKIE);  
?>
```

### ■ Brisanje piškotka

```
<?php  
    setcookie("uporabnik", "", time() - 3600);  
?>
```

## □ Seje

- Hranjenje poljubnih podatkov določenem uporabniku, ki so na voljo v vseh php datotekah

- Začetek seje

- `<?php session_start(); ?>`

- Zapis podatkov v sejno spremenljivko

```
<?php
    session_start();
    $_SESSION["uporabnik"]="janez";
?>
```

- Konec seje

```
unset($_SESSION["uporabnik"]);
...
session_destroy();
```

- SSI – Server Side Includes
  - Vključevanje zunanjih PHP skript oz. združevanje različnih datotek
  - Modularna zgradba spletnih strani
  - Ukaza
    - `include()`
      - opozorilo o napakah
    - `require()`
      - prekinitev izvajanja skripte

- Napake med izvajanjem skript
  - Prekinitev izvajanja s pomočjo funkcije **die( )**

```
<?php
    if(!file_exists("opis.txt"))
    {
        die("Datoteka ne obstaja");
    }
    else
    {
        $file=fopen("opis.txt","r");
    }
?>
```

- Drugi načini prestrezanja napak
  - “Custom error handler”: sami določimo funkcijo, ki se pokliče, ko nastopi napaka

```
error_function(error_level,error_message,  
error_file,error_line,error_context)
```

```
function mojaNapaka($st_napake, $opis)  
{  
    echo "<b>Napaka:</b> [" . $st_napake . "] " . $opis . "</>";  
    echo "Konec skripte";  
    die();  
}
```

```
set_error_handler(" mojaNapaka ");
```



# Podatkovne zbirke

- Podatkovna zbirka – baza podatkov
  - Podatki -> informacije -> znanje
  - Strukturirana zbirka podatkov na nekem računalniku ali strežniku
    - Zbirka podatkov
    - Metapodatki
  - Nadzorovani dostop, branje in pisanje
    - DMBS (Database Management System)

- Vrste podatkovnih zbirk glede na število uporabnikov
  - En uporabnik (single-user)
  - Več uporabnikov (multiuser)
    - Workgroup ( < 50)
    - Enterprise ( > 50)
- Vrste podatkovnih zbirk glede na lokacijo
  - Centralizirane
  - Porazdeljene

- Arhitektura podatkovne zbirke
  - Strojna oprema
  - Programska oprema
    - Operacijski sistem, DBMS, pomožni programi (GUI)
  - Uporabniki
    - Administratorji, načrtovalci, programerji, uporabniki
  - Postopki oz. pravila uporabe zbirke
  - Podatki

- Redundanca (redundancy)
- Nedoslednost (inconsistency)
- Podatkovne anomalije (anomaly)
  - Popravljanje, vpisovanje ali brisanje
- Integriteta podatkov (integrity)
  - Točnost (ni nekonsistence)
  - Preverljivost oz. ponovljivost

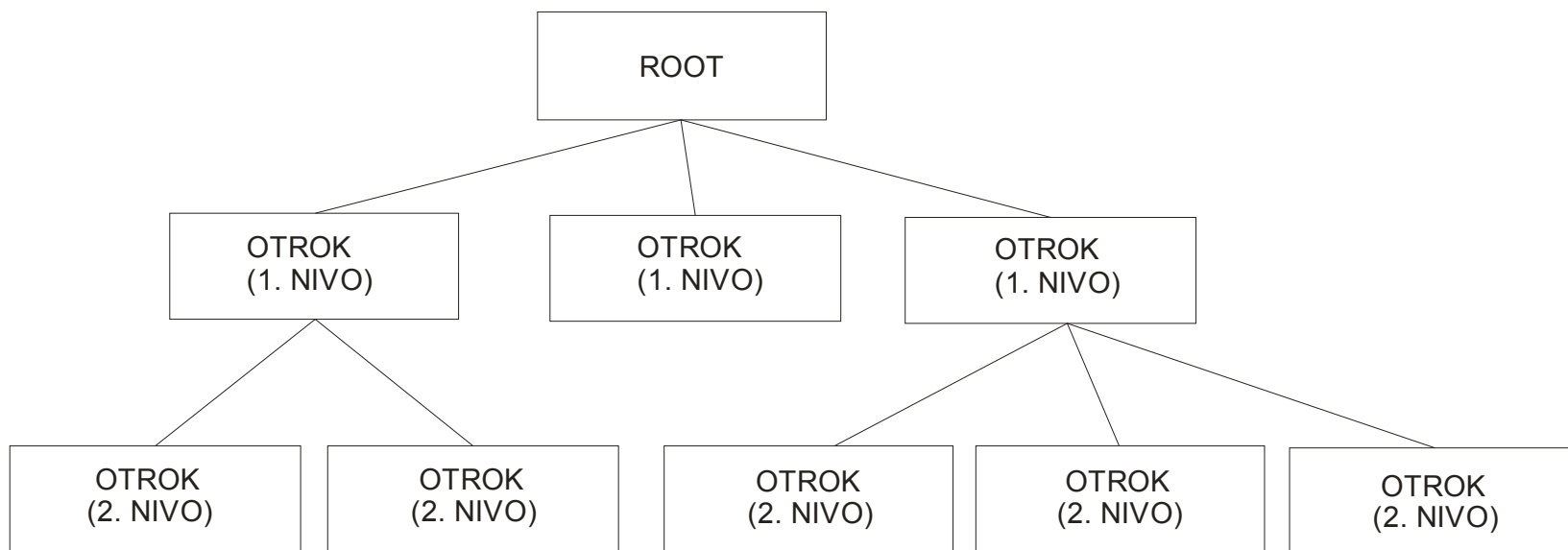


- Osnovni gradniki podatkovni modelov
  - Entiteta
  - Atributi (lastnosti)
  - Relacije (povezave)
    - One-to-many (1:\*)
    - Many-to-many (\*:\*)
    - One-to-one (1:1)
  - Omejitve

- Modeli podatkovnih zbirk
  - Sistem map (file system)
  - Hierarhični model
  - Mrežni model
  - Relacijski model
  - Objektno orientiran relacijski model
  - XML

## □ Hierarhični model

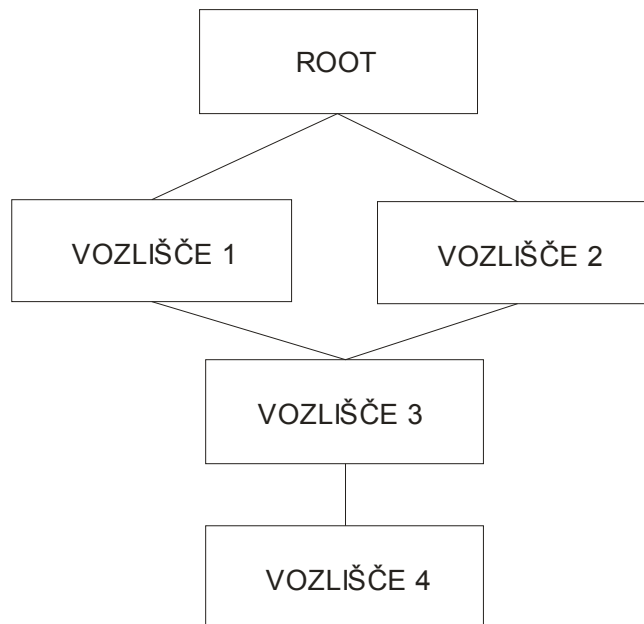
- Tipična drevesna struktura (koren, vozlišča in veje)
  - Relacija one-to-many
    - Vsako vozlišče-starš (parent) ima lahko VEČ pod-vozlišč (child)
    - Vsak otrok ima lahko SAMO ENEGA starša
- Otroci in starši so med seboj povezani z vejami (kazalci)
- Vsak starš vsebuje kazalce do svojih otrok





## □ Mrežni model

- Razširjeni hierarhični model
- Vsak starš lahko vsebuje več otrok in vsak otrok ima lahko več staršev
  - Relacija many-to-many
- Relacije med vozlišči so predstavljene s polji (set)





# Osnove podatkovnih zbirk

- Relacijski model
  - RDBMS (Relational Database Management System)
  - Osnovna enota je **tabela** (table)
    - Zaključena enota za hranjenje podatkov
    - Unikatno ime znotraj podatkovne baze
    - Podatki so organizirani v zapise/vrstice (**records**) in polja/stolpce (**fields**)
    - Logična struktura: dejanska lokacija shranjene tabele je nepomembna
  - Ključi
  - Relacije (povezave)
    - 1:1, 1:\*, \*:\*
  - Relacijski diagram
  - Indeksi

- Manipulacija podatkov preko poizvedb (query-jev)
  - Poizvedbe preko RDBMS
  - Določanje kriterijev
    - Primer: “**Poišči vse zapise v tabeli *Politiki* kjer polje *Ime* vsebuje *Janez***”

Rezultat:	001	Janez Podobnik	SLS
	007	Janez Bond	CIA
	011	Janez Drobnič	NSI

- Structured Query Language
  - Standardiziran jezik za komunikacijo z relacijskimi bazami
- Osnovne funkcionalnosti jezika SQL
  - Pridobivanje zapisov iz baze
  - Vstavljanje in zapisovanje zapisov v bazo
  - Posodabljanje zapisov v bazi
  - Brisanje zapisov iz baze
  - Ustvarjanje novih baz
  - Ustvarjanje novih tabel v bazi
  - ...

- Osnovna SQL sintaksa
  - Spreminjanje vsebine zbirke (DML)
    - SELECT
    - UPDATE
    - DELETE
    - INSERT INTO
  - Spreminjanje strukture zbirke (DDL)
    - CREATE DATABASE
    - CREATE TABLE
    - DROP TABLE
    - CREATE INDEX
    - ...

- **SELECT stavek**
  - Iskanje podatkov v zbirki glede na podane kriterije

**SELECT Ime FROM Osebe**

**SELECT \* FROM Osebe**

**SELECT \* FROM Osebe WHERE Ime="Janez"**



- SELECT stavek
  - FROM: ciljna tabela(e)
  - Primerjave v WHERE stavku
    - =, <>
    - < , >
    - >=, <=
    - BETWEEN, LIKE, IN
  - Logične primerjave: AND in OR

```
SELECT * FROM Osebe  
WHERE (Ime="Janez" AND Priimek<>"Drobnič")
```

## □ SELECT stavek

```
SELECT * FROM Osebe  
WHERE Ime LIKE "%ane%"
```

```
SELECT * FROM Osebe  
WHERE Starost BETWEEN 13 AND 23
```

```
SELECT Priimek FROM Osebe  
WHERE Ime = "Janez"
```



- Združevanje rezultatov

- COUNT, MIN, MAX, SUM, AVG

- Sortiranje zapisov

- ORDER BY, GROUP BY

```
SELECT * FROM Osebe  
WHERE (Ime="Janez" AND Priimek<>"Drobnič")  
ORDER BY Priimek
```

- Združevanje tabel

- JOIN, LEFT JOIN, RIGHT JOIN

```
SELECT Osebe.Ime, Podjetja.Ime  
FROM Osebe LEFT JOIN Podjetja  
ON Osebe.ID_podjetja = Podjetje.ID
```

## □ INSERT stavek

```
INSERT INTO Osebe  
VALUES ("Jaka", "Lakovič", ...)
```

```
INSERT INTO Osebe (Ime, Priimek)  
VALUES ("Manca", "Špik")
```

- UPDATE stavek

UPDATE Osebe

SET Ime="Jakec"

WHERE Ime="Jaka" AND Priimek="Lakovič"

- DELETE stavek

```
DELETE FROM Osebe  
WHERE Priimek="Drobjnak"
```

```
DELETE FROM Osebe
```

- ❑ SQL je “močan” skriptni jezik
- ❑ Možnosti povezovanja tabel
  - INNER JOIN, LEFT JOIN, FULL JOIN, SELECT INTO, itd.
- ❑ Funkcije za delo s spremenljivkami oz. tabelami
  - avg(), count(), first(), last(), sum(), itd.
  - ucase(), lcase(), mid(), len(), itd.

## □ MySQL

- Najpopularnejša odprtokodna podatkovna baza
- RDBMS
  - Ni uporabniškega vmesnika!!
  - Dodatna orodja za administracijo
    - phpMyAdmin
- Deluje na skoraj vseh operacijskih sistemih (Unix, Linux, Mac OS, Windows, itd.)
- Del inštalacije XAMPP paketa
- Ni popolnoma skladna s standardno SQL sintakso

- PHP upravlja z MySQL bazo s pomočjo jezika SQL
- Vzpostavljanje povezave

```
$povezava = mysql_connect(servername,username,password);
```

```
mysql_select_db("ime_baze", $povezava);
```

- Rušenje povezave

```
mysql_close($povezava);
```

## □ Primeri poizvedb

```
$povezava = mysql_connect("localhost","jaka","ggg111");  
mysql_select_db("ime_baze", $povezava);
```

```
$rezultat= mysql_query("SELECT * FROM Avtomobili");
```

```
mysql_query("INSERT INTO Avtomobili(Znamka, Model, Letnik)  
VALUES ('Renault', 'Clio', 2009)");
```

```
mysql_query("UPDATE Avtomobili SET Letnik= 2010  
WHERE Znamka= 'Renault' AND Model = 'Twingo'");
```

```
mysql_close($povezava);
```



## □ Primer obdelave vrnjenih podatkov

```
$povezava = mysql_connect("localhost","jaka","ggg111");  
mysql_select_db("ime_baze", $povezava);
```

```
$rezultat= mysql_query("SELECT * FROM Avtomobili");
```

```
while($row = mysql_fetch_array($rezultat))  
{  
    echo $row['Znamka'] . " " . $row['Model'];  
    echo "<br />";  
}
```

```
mysql_close($povezava);
```

- Open Database Connectivity – ODBC
  - Zagotavlja standardni vmesnik za dostop do RELACIJSKIH podatkovnih zbirk
  - Vmesni sloj med aplikacijo in gonilnikom (driverjem) podatkovne zbirke
  - Neodvisen od OS, vrste podatkovne zbirke in programskega jezika aplikacije
  - Temelji na jeziku SQL (Structured Query Language) -
  - Podpora velikemu številu vrst podatkovnih zbirk

## □ PHP in ODBC

```
$povezava =odbc_connect('moja_baza',"");  
$sql="SELECT * FROM Avtomobili";  
$zapis=odbc_exec($ povezava,$sql);
```

```
while (odbc_fetch_row($zapis))  
{  
    $znamka=odbc_result($zapis,"Znamka");  
    $model=odbc_result($zapis,"model");  
    echo "<tr><td>$znamka</td>";  
    echo "<td>$model</td></tr>";  
}
```

```
odbc_close($zapis);
```

# Primerjava ASP in PHP skriptnih jezikov

- Cena
  - ASP – Potrebna MS licenca zaradi IIS
  - PHP – brezplačno (Linux)
- Hitrost
  - PHP se v povprečju izvaja hitreje kot ASP
- Okolje (platforma)
  - ASP – V večini primerov MS okolje (Linux + Apache)
  - PHP – Linux, Windows, Unix
- Povezovanje s podatkovno bazo
  - ASP – MS SQL
  - PHP - MySQL

## □ ASP datoteka

- HTML datoteka, kateri so dodani segmenti VBScript kode
- Tipična končnica .asp

## □ ASP sintaksa

- ASP datoteka je zelo podobna HTML datoteki, le da vsebuje t.i. VB skripte
- Posamezne skripte so označene z `<% ... %>`  
`<script language="vbscript" runat="server"> .... </script>`
- Komentarji: `'`
- Skriptni jeziki
  - VBScript (privzeti jezik)
  - JavaScript
  - PERL, REXX, itd.

## □ Pošiljanje vsebine odjemalcu

```
<html>
```

```
  <body>
```

```
    <%
```

```
      response.write("Lep pozdrav")
```

```
    %>
```

```
  </body>
```

```
</ html >
```

## □ Skrajšana oblika

```
...
```

```
<%= "Lep pozdrav" %>
```

```
...
```

## □ Najava spremenljivke (**brez tipa !!**)

<%

Dim starost

Dim i

starost = "65 let"

i = 0

%>

<%

Const pi=3,14

%>

<%

dim prijatelji(4)

prijatelji(1)="Miha"

prijatelji(2)="Bojan"

prijatelji(3)="Marjan"

prijatelji(4)="Aleks"

%>



## □ Globalna in lokalna spremenljivka

<%

Dim letalo

Sub Vozila(Tip)

Dim vozilo

vozilo = "Golf"

letalo = "AIRBUS"

End Sub

%>

## □ Procedure in funkcije

```
<%
```

```
Sub seštej_stevili(stevilo1, stevilo2)  
    response.write(stevilo1 + stevilo2)
```

```
End Sub
```

```
%>
```

```
<%call seštej_stevili(3,5)%>
```

```
<%
```

```
Function zdruzi_besedi(beseda1,beseda2)  
    zdruzi_besedi=beseda1 & beseda2
```

```
End Function
```

```
%>
```

```
<%=zdruzi_besedi ("Lep","Dan")%>
```

## □ If, For in Case stavki

If vrednost = 12 Then

...

End if

For i=0 to 10

...

Next

Select Case (vrednost)

Case 1

....

Case 2

...

Case Else

...

End Select

# CGI – Common Gateway Interface

- Program, ki teče na internetnem strežniku
- Vmesnik med zunanjimi aplikacijami in internetnim strežnikom
- Prilagajanje vsebine glede na zahteve odjemalca
  - Procesiranje vhodnih podatkov odjemalca
  - Dostop do podatkovnih baz na strežniku
  - Ustvarjanje vsebine v realnem času (on the fly)
  - Nadzor nad dostopom in pravicami odjemalcev

- CGI programi
  - Napisani v poljubnem programskem jeziku (C, C++, Perl, itd.)
    - Skriptni ali prevedeni jeziki
      - Hitrost in zanesljivost programskih jezikov
      - Berljivost kode in odpravljanje napak
      - Prenosljivost programov
  - Sprejemajo vhodne podatke od internetnega strežnika
    - Podpora standardnim vhodno-izhodnim tokovom: STDIN in STDOUT
  - Izhodni podatki se posredujejo direktno odjemalcu

# Common Gateway Interface

- Okoljske spremenljivke (“environment variables”)
  - Posredovanje podatkov o zahtevi ustreznemu CGI programu
  - Strežnik nastavi vrednosti teh spremenljivk, ko odjemalec pokliče določeno CGI skripto
  - SERVER\_NAME
  - SERVER\_PORT
  - REQUEST\_METHOD
  - QUERY\_STRING
  - REMOTE\_HOST
  - REMOTE\_ADDR
  - CONTENT\_LENGTH

# Common Gateway Interface

- Sprejemanje podatkov od odjemalca
  - S pomočjo metod:
    - GET: podatki so del URL naslova in sledijo znaku “?”
    - POST: podatki so dodani na konec glave HTTP zahteve
  - GET
    - Podatki se shranijo v spremenljivko QUERY\_STRING
  - POST
    - Podatki se shranijo v običajni vhodni tok STDIN
  - Sintaksa shranjenih podatkov
    - ime1=vrednost1&ime2=vrednost2*
  - CGI program vse podatke sprejme kot en niz (string)



- Primer klica CGI programa
  - S pomočjo metode POST

```
<form method="post" action="http://moj_streznik.com/cgi-bin/Obdelaj_obrazec.cgi">  
  <div>  
    <label>Ime: <input type="text" name="ime" /></label>  
    <label>Priimek: <input type="text" name="priimek" value="Sodnik" /></label>  
    ...  
</form>
```

- S pomočjo metode GET

```
<a href="http://moj_streznik.com/cgi-bin/  
Obdelaj_obrazec.cgi?Ime=Jaka&Priimek=Sodnik">Obdelaj</a>
```

- Sprejemanje podatkov od odjemalca
  - POZOR: posebni znaki v zahtevi so zapisani s pomočjo ASCII kode: <, >, ", ', {, }, |, \, ^, ~, [, ], `
  - Posebni znaki: %XX
  - Presledek: +
  - Primer zapisa podatkov:

Ploščina kroga se izračuna kot  $\pi * r^2$ .

Ploščina+kroga+se+izračuna+kot+ $\pi * r^2$

- V QUERY\_STRING oz. STDIN se nahajajo podatki v kodirani obliki

- Stvari, na katere je potrebno paziti
  - Preverjanje pravilnosti oz. smiselnosti vhodnih podatkov
  - Skrivanje informacij o programih in servisih na strežniku
  - Onemogočanje uporabe tehnologije Server Side Includes (SSI)

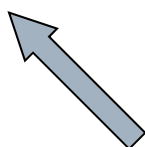
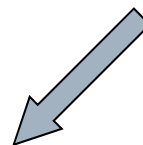
**<!--#exec cgi="/cgi-bin/format.cgi" -->**
  - Možnost zlonamerne uporabe programov na strežniku

- Pošiljanje odgovora odjemalcu
  - Strežnik odgovori odjemalcu preko standardnega izhodnega toka STDOUT
  - Odgovor je lahko HTML dokument, slika, povezava na nek drug dokument, itd.
  - V odgovoru je potrebno definirati tip dokumenta, ki je vsebovan v odgovoru

Primer HTML dokumenta:

**Content-type: text/html\n\n**

POZOR: prazna vrstica



POZOR: Velike in male črke

- V odgovoru odjemalcu moramo v celoti določiti vrnjeno HTML vsebino

## PRIMER:

```
printf("Content-type: text/html\n\n");  
printf("<html><body>");  
printf("<p>Tole je preprost CGI odgovor!</p>\n");  
printf(" </body></html>\n");
```

- Drugi tipi vrnutenih dokumentov:

Content-type: image/gif\n\n

Location: http://www.drug\_streznik.com\n\n

# Common Gateway Interface

## Primer CGI programa

```
#include <iostream.h>
#include <stdlib.h>

void main()
{
    char* lpszRemoteHost = getenv("REMOTE_HOST");

    cout << "Content-type: text/html" << endl << endl
    << "<html>" << endl
    << "<body>" << endl
    << "<p>" < endl
    << "Hello, "
    << lpszRemoteHost
    << "!" << endl
    << "</p>" << endl
    << "</body>" << endl
    << "</html>";
}
```

## Klic CGI programa

```
<form method="post"
action="http://path.to/my/cgi/program">
...
```