

Dynamické www stránky, systémy řízení báze dat a prostředky pro vývoj www aplikací

Systémy www aplikací

WWW (World Wide Web) je dnes společně s elektronickou poštou nejvyužívanější službou internetu. URL (Uniform Resource Locator) slouží ke kompletní adresaci informace na internetu. Udává jak protokol, který je k přístupu využit, tak i **adresu serveru** (doménovou nebo IP) a přesné umístění dokumentu.

Obecné schéma: protocol://[login_name[:heslo]@]adresa_serveru[:port][cesta]

Základním protokolem pro www je http (HypeText Transfer Protocol). Slouží k přenosu HTML dokumentů. HTTP protokol pracuje formou klient-server. Klient zasílá požadavky a server na ně reaguje odpovědí. HTTP existuje v několika verzích 0.9, 1.0 a 1.1. Jsou specifikovány RFC (Request for Comments), které stanoví normy pro protokoly a systémy. HTTP 1.0 – RFC 1945, HTTP 1.1 – RFC 2068.

Pro provoz HTTP a tedy zpracovávání požadavků od klienta je potřeba mít nainstalovaný **http server**. Severů existuje celá řada. Jako nejvhodnější se mi jeví Apache [1], který je dostupný zdarma formou open-source a pro více platforem (v Linuxu bývá obvykle součástí většiny distribucí). Bylo by ale možné využít i např. MS IIS (Internet Information Server). Apache HTTP server je jedním z nejpoužívanějších serverů na internetu. Podle [1] ho využívá až 70% webových stránek na internetu. Apache http server je produktem Apache Software Foundation, která se zabývá open-source softwarovými projekty. Apache se neustále vyvíjí a podporuje nové vlastnosti.

HTTP server zpracovává požadavky klientů, kteří se k němu připojují za účelem získání HTML stránek.

Dynamické WWW stránky

Využití některé verze HTML je sice základem pro www aplikaci, ale pokud chceme, aby stránky aktivně prováděly výpočty či vyhodnocování, aby spolupracovaly s databází a vytvářely kvalitní rozhraní celé aplikace, musíme použít další nástroje.

První možností je použití skriptů a některého skriptovacího jazyka. Jsou to především CGI (Common Gateway Interface) skripty zpracovávané **na straně serveru**. CGI skripty jsou nezávislé na programovacím jazyku. Využití CGI skriptů je dnes výhodné jen za určitých podmínek (dlouhé a složité výpočty a jen krátké zobrazovací výpisy). Další možnostmi jsou JavaScript a VBScript zpracovávané **na straně klientského počítače**.

JavaScript je jazyk interpretovaný prohlížečem a není tedy zpracováván na straně serveru. Díky tomu může být využit k účelům, kdy by bylo využití například PHP krajně nevhodné, například k aktivní stylizaci dokumentů (tvorbě různých menu). Je ale možné ho využít i k jednoduchým výpočtům, interakci s dokumentem či řízení prohlížeče. Jazyk podporuje většinu obvyklých příkazů, výrazů i operátorů, netypané proměnné a využívá objektových vlastností. Do dokumentu se zapisuje tagem `<script language="javascript"> . . . </script>`.

Mezi modernější jazyky zpracovávané serverem patří ASP (Active Server Pages) a PHP (PHP Hypertext Preprocessor). ASP je skriptovací jazyk z dílny firmy Microsoft. Funguje na principu vkládání vsuvek a příkazů serverem a jejich interpretaci. Vlastní ASP se píše za pomoci VBScriptu nebo JScriptu (implementace JavaScriptu od Microsoftu). Novější možností je využití ASP.NET. Nejedná se již o skriptovací jazyk, ale o plnohodnotný programovací jazyk využívající jádra .NET framework. ASP i ASP.NET je komerční aplikací.

Podobné možnosti jako ASP nabízí i PHP. **PHP** je na rozdíl od ASP open source [19], takže jsou dostupné i kompletní zdrojové kódy a je k dispozici na více platformách. Opět pracuje na principu vkládání částí kódu do HTML stránek. Tento kód interpretuje server a musí tedy na něm být podpora instalována. Klient, který výslednou stránku zobrazuje, nepotřebuje žádnou dodatečnou instalaci aplikace nebo nastavení. Klientu (www prohlížeči) je prakticky jedno zobrazuje-li stránku generovanou pomocí ASP, PHP nebo jen statickou HTML stránku. PHP podporuje celou řadu databázových SQL serverů, se kterými dokáže pomocí vlastní sady funkcí komunikovat. Pro administraci MySQL databáze pomocí PHP existují dokonce zvláštní nástroje jako je phpMyAdmin [24].

Stránky obsahující PHP musí být na serveru rozpoznány (například díky příponě php) a potom může být kód zpracován. V praxi se v jednom dokumentu často kombinuje PHP kód spolu s HTML. K oddělení se PHP zapisuje do zvláštních značek:

- `<? . . . ?>`
- `<?php . . . ?>`
- `<SCRIPT LANGUAGE="php"> . . . </SCRIPT>`

Obecně lze využít všech třech způsobů s tím rozdílem, že druhá varianta je vyžadována normou XML.

PHP podporuje proměnné různých typů včetně složených polí a objektů i příkazy typu if, while a podobně. Samozřejmostí je celá řada využitelných funkcí pro matematické výpočty, spolupráci s databázemi, soubory, protokoly a mnohé další [19]. Jeho použití je do jisté míry podobné jazykům C/C++.

Systemy řízení báze dat

Databázi rozumíme uspořádanou množinu informací (dat) uloženou na paměťovém médiu. Systém řízení báze dat (SŘBD) je databáze včetně softwarových prostředků umožňující přístup k těmto datům a manipulaci s nimi. Obvykle je pro jednoduchost pojmem databáze myšlen celý SŘBD.

Databáze slouží k přehlednému ukládání většího množství dat a snadné a efektivní další práci s nimi. Z hlediska ukládání dat a vazeb mezi nimi můžeme rozdělit databáze na následující typy:

- Síťové databáze
- Hierarchické databáze
- **Relační databáze**
- Objektové databáze
- Objektové relační databáze
- Ostatní typy

Nejrozšířenějším typem jsou databáze relační (případně objektové relační). Relační databáze je založena na relačním matematickém modelu. Ze dvou množin dat je možné vytvořit kartézský součin a relace je podmnožinou tohoto součinu odpovídající jisté skutečnosti. Data jsou uložena do tabulek (n-tic) s pevně stanovenými typy položek (sloupce) a jednotlivé položky jsou uloženy v řádcích. Mezi tabulkami majícími stejný typ sloupce mohou pak existovat vazby (relace).

Jazykem pro ovládání databáze je SQL (Structured Query Language - strukturovaný dotazovací jazyk). SQL se však prosadil jako de facto standard a ANSI (American National Standards Institute) založil standard na tomto jazyku. S jazykem se pracuje pomocí sady příkazů pro definici, manipulaci a řízení dat.

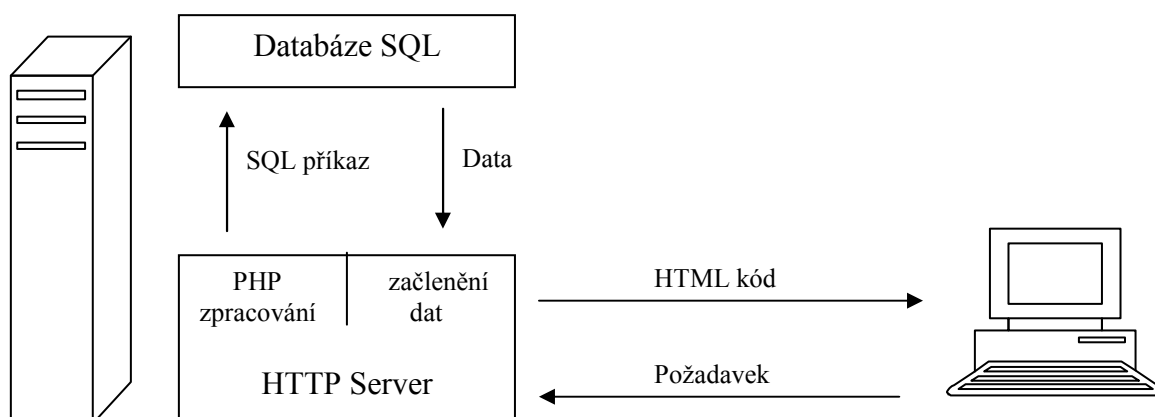
Jazyk SQL využívá celá řada databázových systémů a určit, který je nejlepší, není prakticky možné. Vždy záleží na konkrétních požadavcích, které od systému očekáváme. Mezi nejznámější databázové systémy patří: Oracle, MS SQL, **MySQL**, PostgreSQL:

Každý databázový systém má svoje klady a zápory. Výběr záleží na konkrétní aplikaci, na použitelnosti ve vybraném implementačním jazyku a platformě.

Princip

Za pomoci PHP (a ostatních jazyků) je možné vytvořit distribuovanou webovou aplikaci. Celý systém webového serveru s podporou PHP a databáze SQL pracuje následujícím způsobem. Klient přes www prohlížeč odesílá na server požadavky, server za pomoci PHP a databáze zpracuje výsledný kód a odesílá uživateli čistý HTML kód (případně včetně CSS a JavaScriptu). Požadavky klienta jsou získávány většinou odesláním HTML formulářů a jejich zpracováním pomocí PHP na straně serveru.

Proces je vidět na následujícím obrázku:



Instalace a zprovoznění www stránek

Jak tedy zprovoznit svoje vlastní www stránky? Existuje několik možností. V první řadě si stránky musím vytvořit (nebo nechat vytvořit). Pokud budu používat některé serverové skripty, jako je PHP musím mít při testování (a to je třeba často) nainstalovanou podporu. Mohu si tedy na svůj počítač nainstalovat http server (např. Apache) a PHP. Pak mohu jednoduše lokálně testovat svoje stránky. Pokud budu chtít využívat i databáze, a pro větší aplikace je to téměř nezbytnost musím nainstalovat i databázový stroj. Ke konfiguraci a vytvoření databáze bych ale měl znát, alespoň základy SQL.

Jakmile mám stránky vytvořeny, chci je vystavit na internetu. To modu udělat více způsoby:

Proměnné:

```
$nazev = .....; // nadefinování jména proměnné a popř. přiřazení hodnoty
```

```
isset(); // zjistí jestli proměnná zadaná jako parametr existuje
```

PHP v odkazech:

```
page01.php?str=detailprojektu&akce=uprava
```

// pokud je použito jako adresa odkazu např. u tagu <a> tak ...

přejde na stránku page01.php

a zároveň vytvoří proměnnou \$_REQUEST['str'], které přiřadí hodnotu „detailprojektu”

atd...

Řízení toku dat:

```
if (PODMÍNKA) { // podmíněný příkaz
```

```
    ...
```

```
}
```

```
else {
```

```
    ....
```

```
}
```

```
while (PODMÍNKA) { ..... } // cyklus while
```

```
for (expr1; expr2; expr3) { ... } // cyklus for
```

```
function NAZEV_FUNKCE(PARAMETRY_ODDELENE_CARKAMI) { //definice funkce
```

```
    ...
```

```
}
```

Zpracování formulářů pomocí PHP:

Příklad: soubor Huml s vytvořeným formulářem

V definici formuláře je akce. Akce značí, čím bude soubor zpracováván, zde php skript souboru vyhodnot.php. Method říká, jakou metodou budou data z formuláře odeslaná, tedy předaná skriptu na zpracování. \$_GET se předává v URL, \$_POST formulářem a \$_REQUEST zahrnuje \$_GET, \$_POST a \$_COOKIE. Každý prvek formuláře je pojmenovaný přes name="".

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

```
<input type="text" name="jmeno">
```

```
....
```

```
</form>
```

V souboru vyhodnot.php pak bude přístup k daným proměnným zprostředkován např. následovně:

```
$_REQUEST['jmeno']
```

Další informace o PHP na www.php.net a na Googlu