



Středoškolská technika 2011

**Setkání a prezentace prací středoškolských studentů na
ČVUT**

Sběr dat z čistírny odpadních vod

Radek Vokoun

Vyšší odborná škola a Střední škola slaboproudé elektrotechniky

Novovysočanská 48/280

190 00 Praha 9



1. Úvod

V posledních letech dochází k nasazení WEBových aplikací v oblastech, kde by v dřívějších obdobích nebyly s ohledem na charakter práce očekávány. Mezi hromadně rozvíjející se oblasti, patří řízení a sledování výrobních nebo servisních procesů a poskytování služeb. Při technologickém řešení procesů se využívá měření, přenosu a následného zpracování dat včetně jejich zobrazení. Data jsou sledována i z hlediska mezních hodnot (tj. pomocí nich je usuzováno, zda systém pracuje v mezích předem navržených parametrů).

Zpracovaná data jsou pak dostupná jednak pro servisní techniky, kteří pomocí nich kontrolují správnou funkci systému a v případě překročení mezních hodnot vyjíždějí na servisní zásah. Dále data sledují i majitelé zařízení, kteří je využívají ke kapacitním rozborům a přehledům jaké množství látek bylo zařízením zpracováno.

Při navrhování systému, byla zajištěna podmínka připojení čistírny k internetu. Data systému mají být přenášena na server, kde mají být ukládána do databáze a následně zobrazována ve webovém prohlížeči.

Aplikace má za úkol:

- přečíst data z databáze čističky, uložit je jako XML soubor a odeslat je na centrálu
- přečíst XML soubor a data uložit jako přírůstek do databáze
- zobrazit a seřadit data a dále je nezpracovávat

Proto jsem aplikaci realizoval jako WEBovou aplikaci, skládající se z následujících částí:

- část software uvnitř čističky
- část příjmu a uložení dat na centrále
- část zobrazení dat

2. Rozbor práce

Za prvé bylo nutné si určit, do jakého typu souboru se budou data z databáze ukládat. Pro svou jednoduchost jsem zvolil jazyk XML (*Extensible Markup Language*, česky *rozšiřitelný značkovací jazyk*). Je to formát dokumentů, který nám dovoluje zapsat data společně s jejich významem. Jako jádro pro konverzi z MySQL do XML jsem použil již vytvořený a volně šiřitelný (pod licencí GNU GPL) XML framework.

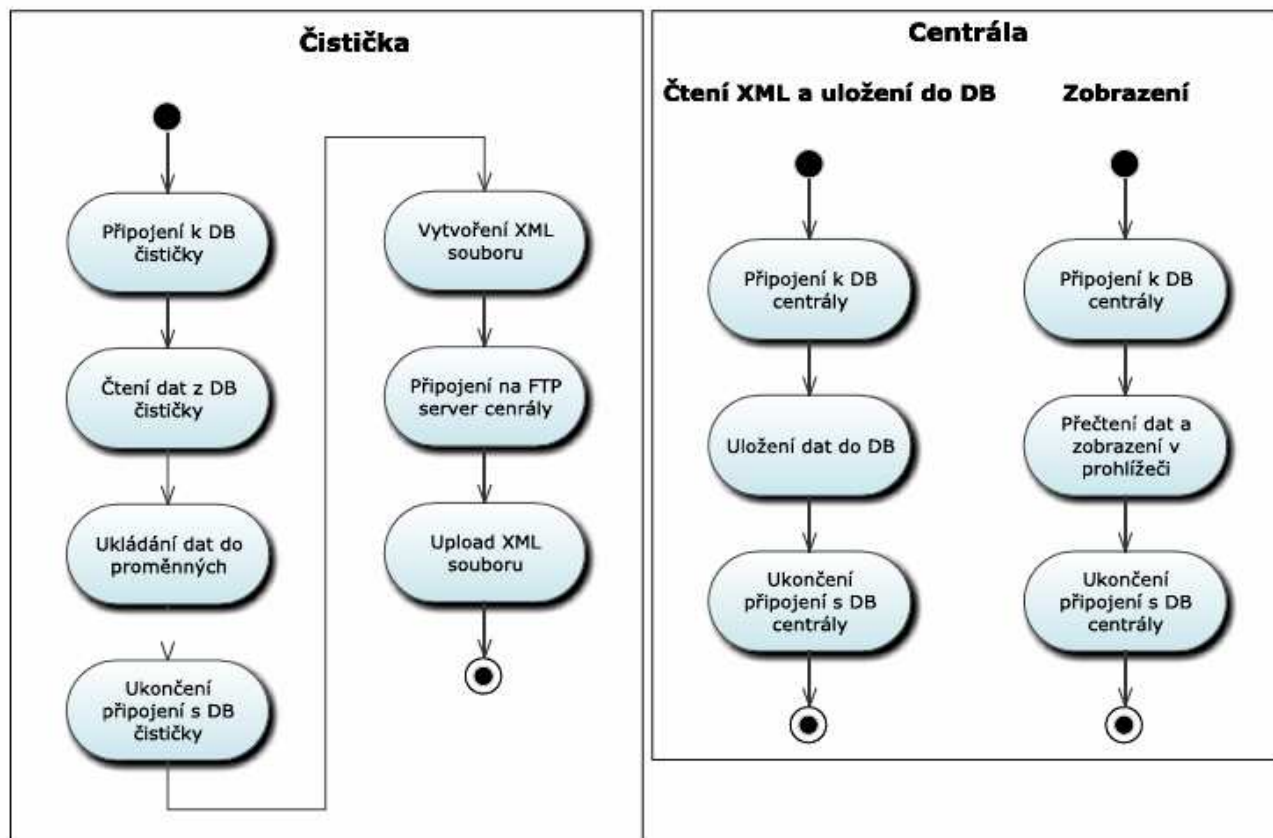
Pro samotný přenos dat jsem zvolil metodu přenosu přes protokol FTP, který je v PHP implicitně podporován a server pro něj je velmi snadno naprogramovatelný.

Pro zobrazení dat z jednotlivých dnů jsem použil open-source JavaScript kalendář a pro jednodušší zobrazení času zbývajcího do dalšího update databáze JavaScriptový odpočet času, který jsem nastavil na 15 minut dle zadání. Datum posledního update se ukládá do databáze a je použito pro výpočet přesného času, kdy dojde k dalšímu update.

Skripty se provádí každých 15 minut pomocí služby CRON (Linux) nebo Plánovače úloh (Windows). Skript pro vkládání dat do databáze, který je na serveru centrály, musí mít čas vykonání posunut alespoň o 1-2 minuty, nejlépe o 5 minut, kvůli možnému zpoždění vykonání skriptu v ČOV (Čistírna odpadních vod). Kdyby se totiž skripty spouštěly ve stejný čas, docházelo by k tomu, že se budou do databáze zapisovat data, která byla přijata v předchozích 15-ti minutách, tzn. data by nebyla aktuální.

Veškeré konfigurační parametry jsou zapsány v souboru „config.php“. Parametry jsou popsány přímo v souboru. Na složku „data“ v části určené pro čistírnu je nutné nastavit přístupová práva ke složce pro zápis, jinak se nebudou data ukládat. 4

1. Blokové schéma aplikace



Obrázek 1

Popis obrázku: V první části obrázku je schéma aplikace umístěné v ČOV. Má za úkol číst data z databáze, ukládat je do XML souboru a odeslat přes FTP do centrály. Druhá část má za úkol přečíst XML soubor, vložit načtená data do databáze a poté je zobrazit ve webovém prohlížeči a seřadit primárně podle data měření a dále řadit podle toho, který sloupec si uživatel vybere. 5

3. Co je to PHP

3.1 Historie jazyka PHP

PHP je skriptovací jazyk pro tvorbu dynamického webu a jeho počátky spadají do roku 1994, kdy se pan Rasmus Lerdorf rozhodl vytvořit jednoduchý systém pro počítání přístupu ke svým stránkám, který byl napsán v PERLu. Za nějakou dobu byl systém přepsán do jazyka C, protože PERL velmi zatěžoval server. Balík těchto skriptů byl ještě ten samý rok vydán pod názvem „Personal Home Page Tools“, zkráceně PHP. Poté se ještě jmenovalo „Personal Home Page Construction Kit“.

3.1.1 PHP/FI

Protože byla zapotřebí větší funkčnost, Rasmus Lerdorf vytvořil mnohem rozsáhlejší implementaci jazyka C, která umožňovala komunikaci s databázemi a zasvěceným uživatelů vytvářet jednoduché dynamické webové aplikace. Tím vzniklo PHP/FI, což znamená „Personal Home Page / Forms Interpreter“ a to již mělo základní funkce, které známe dnes. Mělo proměnné podobné PERLu, možnost vkládání HTML kódu, zpracování formulářů a syntaxí velice podobnou PERLu, ale bylo oproti PERLu funkčně mnohem limitovanější a jednodušší. Lerdorf veřejně vydal PHP 8. června 1995, aby mohl najít co nejvíce chyb a tak zdokonalit kód.

V roce 1997, kdy byla vydána beta verze PHP/FI 2.0, mělo již PHP několik tisíc uživatelů po celém světě se zhruba 50 000 instalacemi na doménách (tj. zhruba 1% všech domén na internetu v té době). V listopadu roku 1997 byla oficiálně vydána finální verze PHP/FI 2.0. Krátce poté byla vydána první alfa verze PHP 3.0.

3.1.2 PHP 3

PHP 3.0 vytvořili Andi Gutmans a Zeev Suraski. Kompletně přepsali předchozí verzi PHP/FI 2.0, které nedostačovalo jejich potřebám k vytvoření internetového obchodu, na kterém pracovali pro jeden Univerzitní projekt. Ve snaze zavedení nové verze mezi již existující skupinu uživatelů, začali Andi, Zeev a Rasmus spolupracovat a označili PHP 3.0 jako oficiálního nástupce PHP/FI 2.0 a vývoj PHP/FI 2.0 byl v podstatě ukončen.

Jednou z největších výhod PHP, byla jeho možnost rozšiřování. Kromě poskytování solidní infrastruktury pro mnoho různých databází, protokolů a API (rozhraní pro 6

programování aplikací), přilákalo PHP 3.0 také spoustu vývojářů, kteří vytvářeli další rozšíření. Toto bylo pravděpodobně klíčem úspěchu PHP 3.0. Dalším klíčovým prvkem, byla podpora syntaxe objektově orientovaného programování a také mnohem silnější a konzistentnější syntaxe jazyka jako takového.

Jazyk byl také přejmenován z PHP/FI, které v názvu obsahovalo limitaci pro osobní využití, na prosté PHP, které znamenalo „PHP: Hypertext preprocessor“.

Koncem roku 1998 se odhadoval počet instalací PHP 3.0 v řádu desítek tisíc u uživatelů a stovek tisíc na webových serverech. Na vrcholu své slávy, bylo PHP 3.0 nainstalováno na zhruba 10% všech webových serverů na internetu.

3.1.3 PHP 4

V zimě roku 1998, krátce poté, co bylo vydáno PHP 3, začali Andi a Zeev znovu přepisovat jádro PHP. Hlavními cíli bylo vylepšení výkonu komplexních aplikací a zlepšení rozšiřitelnosti základního kódu. Aplikace psané v PHP 3, které pracovaly s novými funkcemi, širokou škálou databází a API třetích stran, byly velice pomalé, protože PHP 3.0 nebylo stavěné na tak náročné aplikace a prostě je nezvládalo.

Nový engine, nazvaný „Zend Engine“ (podle křestních jmen vývojářů – Zeev a Andi) splňoval cíle návrhu a byl poprvé představen v polovině roku 1999. Nové PHP 4, založené na tomto enginu, bylo doplněno spoustou nových funkcí a bylo oficiálně uvolněno v květnu 2000, skoro 2 roky od uvolnění PHP 3. Kromě vysoce zvýšeného výkonu, dostalo PHP 4 spoustu nových funkcí, jako je podpora pro mnoho webových serverů, bezpečnější zpracování vstupů uživatele a HTTP sessions.

Tato verze byla používána odhadem stovkou tisíc uživatelů a několika milióny serverů, což představovalo zhruba 20% všech domén na internetu.

3.1.4 PHP 5

PHP 5 bylo vypuštěno v červenci 2004 po dlouhém vývoji a několika pre-verzích. PHP 5 obsahuje nový Zend Engine 2.0, který obsahuje nový vylepšený objektový model a velké množství nových nebo vylepšených funkcí, jako např. přepsaná podpora XML.

Zatím je PHP 5 stále ve vývoji ve smyslu opravování bezpečnostních a funkčnostních chyb, ale zároveň se již od roku 2006 vyvíjí nové PHP 6, kterému budou odebrány některé funkce PHP 5, ale na druhou stranu jich hodně přibude. 7

4. Proč ano a proč ne

4.1 Proč je PHP tak oblíbené?

K tomu vede řada věcí:

- PHP je relativně jednoduché na pochopení
- PHP má syntaxi velmi podobnou jazyku C a je tedy většině vývojářů dost blízký
- PHP podporuje širokou řadu souvisejících technologií, formátů a standardů
- je to otevřený projekt s rozsáhlou podporou komunity
- dá se najít velké množství již hotového kódu k okamžitému použití nebo funkční PHP aplikace.

- podstatná část z hotového kódu je šířena pod svobodnou licencí „PHP License v3.01“ a dá se použít ve vlastních projektech

- PHP si dobře rozumí s webovým serverem Apache (aby ne, vždyť je to sesterský projekt spravovaný Apache software foundation)

- PHP snadno komunikuje s databázemi, jako je MySQL, PostgreSQL a řada dalších

- PHP je multiplatformní a lze jej provozovat s většinou webových serverů a na většině dnes existujících operačních systémů

- většina dnešních poskytovatelů webových služeb podporuje PHP

4.2 Má PHP také svoje nevýhody?

Víceméně nemá, pro psaní dynamického webu je PHP vždy dobrou volbou, ale je zde pár výhod / nevýhod:

- PHP je interpretovaný (např. jako HTML), ne kompilovaný jazyk (např. jako C)

- kdokoli má přímý přístup k serveru, může nahlédnout do vašich PHP skriptů

- podpora objektového programování není v PHP na moc dobré úrovni. V PHP 5 se to již zlepšilo.

- protože je PHP aktivně vyvíjen, v budoucích verzích jazyka se mohou některé funkce změnit nebo se mohou chovat jinak než dosud.

8

4.3 Co se v PHP dá napsat?

Téměř všechno, co nějak souvisí s dynamickým webem a/nebo s databázemi. Mezi nejčastější aplikace psané v PHP patří například:

- internetové obchody (Zen Cart, VirtueMart)
- podnikové informační systémy (ať už intranetové nebo internetové)
- diskusní fóra (phpBB)
- redakční systémy (Joomla!, Drupal, Wordpress)
- firemní prezentace
- dynamické osobní stránky
- webových poštovních či databázových klientů (SquirrelMail)
- vyhledávače a katalogy (Centrum.cz)
- drobnosti typu počítadla, ankety a mnoho dalších

5 Jak PHP funguje

5.1 Princip PHP

V počátcích internetu byly všechny webové stránky statické. Tak, jak byla stránka napsána, tak byla odeslána do prohlížeče a byla také tak zobrazena. Postupem času, když

už přestávaly statické stránky stačit, vyvíjela se celá řada technologií, které měly stránky rozpohybovat. Tyto technologie se dají rozdělit do dvou skupin a to na „klientské“ a „serverové“.

„Klientské“ technologie jsou založeny na principu odesílání částí kódu společně s HTML stránkou. Takovýto kód se ve vhodné chvíli spustí v klientském prohlížeči, např. při najetí nebo kliknutí myši na odkaz, načtení stránky apod. Spouštění v klientském prohlížeči může být také nevýhoda, protože prohlížeč musí znát programovací jazyk, ve kterém je kód napsán, popřípadě si uživatel musí doinstalovat obslužný program pro daný jazyk, např. Java. Klasickým příkladem „klientské“ technologie je Javascript. 9

"Serverové" technologie jsou založeny na jiném principu. Při vyžádání webové stránky prohlížečem dojde k tomu, že server nejprve zpracuje daný kód, sestaví stránku a poté ji odešle zpět prohlížeči. Právě PHP je technologie běžící na serveru. Klasické PHP obsahuje kusy HTML kódu a hlavně kusy programového kódu. Když server obdrží požadavek na zpracování takového skriptu, tak vezme části HTML kódu a nechá je tak jak jsou, následně vezme části programového kódu, které provede a výsledek zkombinuje s HTML kódem a odešle výsledek prohlížeči.

Tento princip má ohromné výhody. Server totiž může provést jednu nebo několik operací a výsledek pošle do prohlížeče jako standartní HTML stránku. Tím pádem odpadají jakékoli požadavky na konfiguraci klientské stanice, vše se provádí na serveru.

Oba principy předvedu na následujících ukázkách. Následující kód je obyčejná HTML stránka. Když si ji prohlížeč vyžádá, uvidíte v něm jen řádek mezi značkami.

```
<html> Já jsem obyčejná stránka. Nic neumím. </html>
```

Následující kód je PHP skript. Skládá se jednak z HTML kódu, a jednak z PHP kódu. Mezi značkami `<? a ?>` je právě PHP kód. Příkaz `echo` vypisuje text na standardní výstup a příkaz `Date` vrací aktuální čas:

```
<html> Já jsem speciální stránka. Vím, že právě je <? echo Date("G:i") ?> </html>
```

Tato stránka vypíše aktuální čas na serveru v době, kdy byl kód zpracováván. To proto, že kód mezi značkami `<? a ?>` se na serveru nejprve provede a teprve pak se výsledek odešle prohlížeči. Do prohlížeče dorazí toto:

```
<html> Já jsem speciální stránka. Vím, že právě je 21:08 </html>
```

Prohlížeč vůbec neví a ani to vědět nemusí, co se všechno na serveru dělo než mu server odeslal výstup skriptu, vidí jen samotný výsledek. Na serveru může probíhat spousta věcí, jako jsou přístupy k databázím, zpracování obrázků, formulářů, apod. 10

Jak již napovídá „název“ této technologie, ke spuštění PHP skriptu je zapotřebí nějakého serveru. Tím může být například server Apache, který je asi nejlepším řešením pro provozování PHP. Server donutíme zpracovat PHP skript tak, že do adresového řádku v prohlížeči zadáme adresu url, např. „http://localhost/index.php“. Tím prohlížeč řekne serveru, aby skript zpracoval a výsledek odeslal zpět do prohlížeče.

PHP skripty vlastně nejsou vůbec závislé na použitém prohlížeči, protože ten s nimi nepříjde do styku. Všechno provádí server a do prohlížeče dorazí jen HTML.

Kusy PHP kódu se do skriptů vkládají do speciálních značek a tím vzniká možnost kombinace HTML a PHP kódu v jednom souboru.

Možnosti kombinací mohou být takové, že celý soubor bude jeden dlouhý PHP kód, který začíná a končí znaky `<?php a ?>` a HTML kód bude vkládán např. pomocí příkazu `echo`, nebo se PHP může vkládat do HTML pomocí značek `<?php a ?>`.

Zde jsou ukázky vkládání HTML a PHP do jednoho souboru:

```
<?php
echo „<html> zde je HTML kód </html>“;
?>
nebo
<html>
Zde je HTML kód
<?php ?>
</html> 11
```

6. MySQL

6.1 Co je to MySQL

MySQL je multiplatformní relační databázový systém. MySQL je open-source projekt pod veřejnou licencí GNU General public licence a zároveň i pod několika vlastnickými dohodami. MySQL bylo dříve vlastněno a sponzorováno společností MySQL AB, ale nyní je vlastněno společností Oracle Corporation.

Spousta projektů založených na volně šiřitelném softwaru vyžaduje plnohodnotný databázový systém, a proto volí MySQL. Samozřejmě je zde i několik placených verzí, které nabízí nějaké ty funkce navíc. Zde jsou příklady některých z volně šiřitelných projektů: Joomla!, WordPress, phpBB, Drupal, Zen Cart.

MySQL je také používáno u některých velice rozsáhlých projektů, jako jsou Wikipedia, Google (tím nemyslím vyhledávání), Facebook, Flickr, Nokia.com a Youtube.

MySQL je také součástí softwarového balíku LAMP. LAMP je zkratkou pro „Linux, Apache, MySQL, PHP/PERL/Python“. MySQL je napsáno v jazyku C a C++. Jeho SQL parser (syntaktický analyzátor) je napsán v jazyce „yacc“ (generátor parserů pro UNIXové operační systémy; zkratka znamená „Yet Another Compiler Compiler“)

6.2 Historie MySQL

Firmu MySQL AB založili v roce 1995 Michael Widenius a Allan Larsson ve Švédsku. V roce 2000 se stalo MySQL open-source projektem. Roku 2001 bylo hlášeno něco okolo

2 milionů instalací MySQL na serverech i na klientských stanicích. Následující rok byl počet aktivních instalací okolo 3 milionů a MySQL AB otevřelo svojí další centrálu v USA. V roce 2006 má společnost 320 zaměstnanců, z toho jich 70% pracuje z domova, a podíl na trhu se zvýšil na 33%. Roku 2008 byla společnost MySQL AB odkoupena společností Sun Microsystems za cenu přibližně \$1 mld. Oba zakladatelé začali veřejně kritizovat Sun a krátce na to oba společnost opustili. V roce 2009 společnosti Sun Microsystems a Oracle oznámili, že uzavřeli definitivní dohodu, že Oracle odkoupí kmenové akcie Sun za \$9.50 za akcii v hotovosti. Celá transakce měla hodnotu přibližně \$7.4 mld. 12

7. Srovnání s ostatními programovacími jazyky

7.1 PHP vs. ASP

PHP a ASP jsou programovací jazyky standardně používané k tvorbě webových stránek.

ASP je zkratkou pro Active Server Pages. Je to program, který pracuje s IIS neboli Internet Information Services od Microsoftu. ASP potřebuje pro běh Microsoft Server. Na druhé straně, PHP neboli Hypertext Preprocessor, funguje na Linuxu, Unixu i Windows.

PHP má syntaxi velmi podobnou C/C++, tudíž je pro programátory pracující s těmito jazyky přívětivější než ASP. ASP je syntaxí velmi podobné Visual Basicu, nejspíše proto, že Visual Basic je také produkt Microsoftu. Když si programátor volí mezi ASP a PHP, zvolí nejspíše jazyk, který má syntaxi podobnou jazyku který zná.

Když přijde na ceny a výdaje, ASP potřebuje k běhu Windows Server s instalovaným IIS, proto musíte zakoupit obojí k provozování ASP aplikací. Na druhé straně PHP potřebuje k běhu pouze Linuxový server s Apachem, které je možno získat zadarmo.

PHP je daleko více flexibilní v oblasti databází. Může se propojit s mnoha databázemi, z nichž nejvíce používaná je MySQL, která je také k dostání zdarma. Ale pokud budete chtít používat ASP, musíte ještě kromě Windowsového serveru zakoupit také MS-SQL, které je také z dílen Microsoftu.

Rychlost načítání stránek a provádění kódu, je další faktor při výběru jazyka. Pokud budete chtít rychlost, budete muset využít PHP. PHP běží mnohem rychleji oproti ASP, protože běží samotné ve svém vlastním prostoru v operační paměti, zatímco ASP běží jako komponenta nadřazeného serveru.

Většina nástrojů používaných buď pro tvorbu, nebo běh programu psaného v PHP je open-source, takže za ně nic neplatíte. U ASP je to ale jinak, zde nejspíše budete muset dokupovat další programy pro práci s programy napsanými v tomto jazyce. 13

7.2 PHP vs. Java

Pokud se zeptáte programátora používajícího jazyk Java, co si myslí o PHP, nejspíše vám odpoví že PHP je velmi základní, neprofesionální jazyk, který může být velmi nepřehledný a neuspořádaný. Když se zeptáte PHP programátora, co si myslí o Javě, řekne vám, že je to velmi složitý a těžkopádný jazyk a vývoj aplikací zabere spoustu času.

Základním rozdílem mezi PHP a Javou, byla podpora OOP (objektově orientované programování). PHP až do verze 4 nepodporovalo OOP. Částečná podpora přišla s verzí 4, velmi dobrá podpora je až od verze 5. Na druhou stranu Java je na OOP postavená.

Většina začínajících webových programátorů volí spíše PHP pro svou jednoduchost a také proto, že podporuje jak OOP, tak i klasické procedurální programování, které je jednodušší na naučení. Na Javu přecházejí až později, když jim PHP přestane stačit nebo chtějí vytvořit složitější projekt, na který se PHP nehodí a na kterém budou třeba pracovat ve větším týmu.

7.3 Závěr PHP vs. Ostatní jazyky

Na závěr bych řekl, že všechny tyto jazyky mají své výhody a nevýhody. Vše záleží na prostředcích programátora a na tom, co se od webové stránky očekává. Budete si nejspíše klást takovéto otázky: Kolik jste ochotni zaplatit za webovou stránku? Jaký programovací jazyk nyní znáte? Chcete co nejstabilnější a nejrychlejší web? Volba mezi nimi záleží především na osobních preferencích. Také není žádný problém si pohovořit s ostatními programátory nebo webmastery na specializovaných fórech, a vyhledat si více informací o tom, který z programovacích jazyků bude lépe splňovat potřeby vaší webové stránky. 14

8 Přílohy

Všechny skripty jsou na přiloženém CD ve složce „skripty“ a jsou rozděleny na dvě části, část pro čistírnu a část pro centrálu. Také je přiložen kompletní manuál PHP ve formátu CHM (pokud používáte Windows XP SP2 nebo novější, musíte nejdříve ve vlastnostech souboru (pravý klik myši na soubor a volba „Vlastnosti“) kliknout na tlačítko „Odblokovat“, jinak soubor nápovědy nebude fungovat správně a bude zobrazovat chybovou hlášku „Navigace na webovou stránku byla zrušena“).