VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

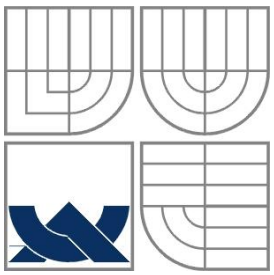
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM HOTELNET

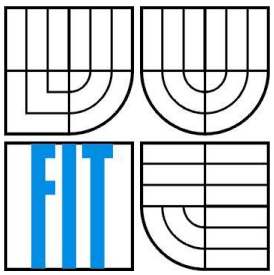
BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE Aleš Šturala
AUTHOR

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM HOTELNET

HOTELNET

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

Aleš Šturala

VEDOUCÍ PRÁCE
SUPERVISOR

Mgr. Masopust Tomáš

BRNO 2007

Abstrakt

Cílem mé práce bylo vytvořit systém ovládaný přes webové rozhraní, který umožňuje administrátorům spravovat uživatele a počítače v malé síti (do 250 počítačů) a spravovat celou tuto síť.

Systém je modulární. Jeho výhodou tedy je, že pokrývá požadavky různých druhů sítí. Vše je ovládáno přes přehledné webové rozhraní. Součástí celého systému jsou PHP skripty pro Linux, běžící na routru, které umí komunikovat se systémem a nastavit router podle údajů zadaných přes webové rozhraní. Uživatelům jsou přiřazena práva, podle kterých mají přístup do některých částí systému.

K vytvoření systému jsem použil v dnešní době běžné technologie pro tvorbu dynamických webových aplikací jako jsou databázový systém PostgreSQL, skriptovací jazyk PHP a značkovací jazyk HMTL.

Klíčová slova

Internet, WWW, informační systém, databázový systém, databáze, PostgreSQL, HTML, PHP, CSS, Apache, Linux, QOS (Quality Of Service), traffic control, HTB, DNS, DHCP, LAN, sítě

Abstract

The main objective of my work was to create a system administered over the web interface, where administrators may control users and computers in a small-area network (max. 250 computers).

The system is modular to meet requirements of different networks and is administered over the web interface. A part of the whole system is created by PHP scripts for a Linux router, which are connected to the database and can set the router. Users have assigned rights to access to different parts of the system.

The system was developed in common technologies like database system PostgreSQL, script language PHP, and mark-up language HTML.

Keywords

Internet, WWW, Information System, Database System, Database, PostgreSQL, HTML, PHP, CSS, Apache, Linux, QOS (Quality Of Service), traffic control, HTB, DNS, DHCP, LAN

Citace

Aleš Šturala: Informační systém Hotelnet, bakalářská práce, Brno, FIT VUT v Brně, 2007

Informační systém Hotelnet

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Mgr. Tomáše Masopusta

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Aleš Šturala
2. 5. 2007

Poděkování

Rád bych tímto poděkoval vedoucímu mé bakalářské práce panu Mgr. Tomáši Masopustovi za odborné vedení.

© Aleš Šturala, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

1. Úvod 8
2. Internet 9
    2.1.1. WWW (World Wide Web) 9
        2.1.1.1. HTTP 9
        2.1.1.2. WWW stránka 9
        2.1.1.3. HTML 9
        2.1.1.4. PHP 10
        2.1.1.5. JavaScript 10
        2.1.1.6. CSS (Cascade Style Sheet) 11
    2.2. Databázový systém 11
        2.2.1. Databázové modely 12
            2.2.1.1. Hierarchický model 12
            2.2.1.2. Síťový model 12
            2.2.1.3. Relační model 12
        2.2.2. PostgreSQL 12
    2.3. Počítačové sítě 13
        2.3.1. Linux 13
        2.3.2. Traffic control 13
3. Realizace 15
    3.1. Specifikace požadavků 15
    3.2. Analýza požadavků 15
    3.3. Databáze 16
    3.4. Systém 17
        3.4.1. Instalace systému 18
            3.4.1.1. Kroky instalace 19
        3.4.2. Instalace a odstranění modulů 20
    3.5. Popis modulů 20
        3.5.1. Admin 20
        3.5.2. Report 24
        3.5.3. FTP search 26
        3.5.4. online 27
    3.6. Skripty 28

3.6.1. traffic control 28
3.6.2. FTP search 30
3.6.3. online 30
4. Závěr 35
5. Použité zdroje 36

1 Úvod

V poslední době došlo k masovému rozšíření amatérských wifi sítí. K vytvoření a administraci takovéto sítě jsou potřeba znalosti systému Linux, routování, traffic control, DNS, DHCP a dalších pokročilých technik.

Systém jsem se rozhodl vyvinout z důvodu absence komplexního systému běžícího na operačním systému Linux, který by pomohl administrovat takovéto sítě. Samozřejmě každá síť vyžaduje odlišné nastavení a má specifické potřeby, a proto je můj systém založen z převážné části na modulech. Administrátor podle typu sítě a podle požadavků na síť vybere moduly, které chce mít v systému, a tyto moduly se nainstalují během instalace celého systému. Celý systém a samozřejmě také moduly a skripty jsou napsány v programovacím jazyku PHP.

2 Technologie

2.1 Internet

Internet je celosvětově rozšířená síť pracující na protokolu TCP/IP, která vznikla z Arpanetu, vládní sítě vytvořené pro potřeby armády.

2.1.1 World Wide Web

WWW je nejpoužívanější služba Internetu, která slouží k přenosu HTML stránek. Tato služba pracuje na principu klient/server, tzn. soubory (HTML, flash, zvuky) jsou uloženy na serveru a klienti se na tyto soubory dotazují.

2.1.1.1 HTTP (Hypertext Transfer Protokol)

Jedná se o patentovaný otevřený internetový protokol, jehož původním účelem bylo poskytnout způsob, jak publikovat a přijímat HTML stránky. Vývoj HTTP protokolu byl koordinován W3C (World Wide Web Consortium).

Tento protokol slouží ke komunikaci mezi WWW serverem a klienty a jedná se o protokol typu request/response (požadavek/odezva). Verze HTTP protokolu jsou 0.9, 1.0 a 1.0. Veze 0.9 nebyla nikdy moc rozšířená. Verze 1.0 je ještě dnes velice používaná, zejména proxy servery. Verze 1.1 je nejnovější verze protokolu HTTP.

2.1.1.2 WWW stránka

WWW stránku bychom mohli přirovnat k prezentaci. Stránka se skládá většinou z textů, obrázků a odkazů, případně flashové aplikace. V poslední době došlo k rozšíření tzv. dynamických WWW stránek - jedná se o stránky, které se dynamicky generují podle uživatelských vstupů (formuláře).

2.1.1.3 HTML

Hypertext Markup Language slouží k tvorbě WWW stránek. K definici a formátování obsahu WWW stránek jsou tzv. HTML tagy, které navíc umožňují vytvářet hypertextové odkazy, které slouží k navigaci v dokumentu nebo mezi WWW stránkami.

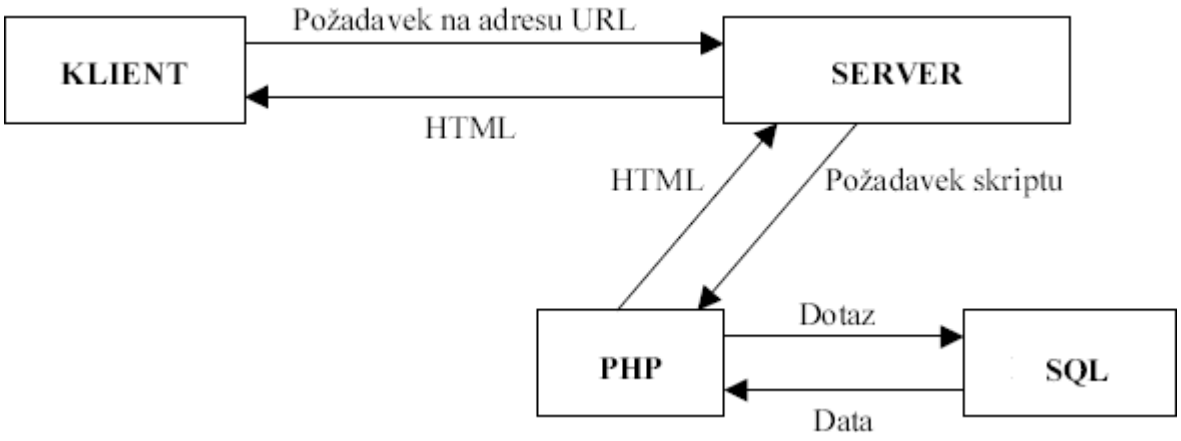
Jazyk HTML byl původně definován panem Timem Bernes-Lee a později společností IETF (Internet Engineering Task Force). Dnes je specifikace jazyka HTML udržována organizací W3C.

2.1.1.4 PHP

Zkratka PHP původně znamenala "Personal Home Page". Tuto technologii vymyslel v roce 1994 Rasmus Lerdorf ke sledování návštěvníků svých stránek. S rozšířením této technologie se ujal název "PHP - Hypertext Preprocesor".

PHP je dnes velmi rozšířená technologie umožňující programování na straně serveru. Jedná se o skriptovací jazyk vkládaný do HTML kódu. Při požadavku na server je stránka nejdříve zpracována PHP interpretem, jsou vykonány všechny příkazy a klientovi odeslán vygenerovaný HTML kód (viz. obrázek 1). Server většinou takto zpracovává soubory s příponou .php, .php3 a .phtml (záleží na konfiguraci daného serveru). Příkazy jsou vkládány přímo do HTML kódu odděleny tagy '<?' a '?>' (případně '<?php' a '?>').

Výhodou PHP je nezávislost na platformě (může běžet ve většině operačních systémů) a jde o serverovou technologii, tzn. že se interpretuje na serveru, nikoliv na klientovi. Mezi další výhody patří výkon, integrace s většinou databázových systémů, stabilita a možnost rozšíření.



Obr. 1. - Komunikace dynamické webové PHP aplikace a SQL databáze

2.1.1.5 JavaScript

Jedná se o klientský skriptovací jazyk, tzn. že se interpretuje u klienta a není potřeba server (není potřeba kompilace). Bohužel JavaScript oproti PHP má řadu nevýhod vyplývajících z faktu, že je interpretován u klienta. Např. z bezpečnostních důvodů neumožňuje číst a ukládat soubory, kdokoliv si může zobrazit zdrojový kód Javascriptu a hlavně různé webové prohlížeče interpretují Javascript různými způsoby, nebo naopak kód vůbec neinterpretují.

2.1.1.6 CSS (Cascade Style Sheet)

V počátcích webu se objevilo velké množství HTML tagů. Různé prohlížeče podporovaly své vlastní tagy a některé další zase neuměly. Nastal tedy problém, že jedna stránka se v různých prohlížečích zobrazovala jinak. Tento problém se snažilo vyřešit World Wide Web Consortium, které v roce 1997 vyvinulo standart CSS, který měl tyto problémy eliminovat a navíc vedl k jednoduchému modelu, který odděluje styl stránky od jeho obsahu. CSS je definice stylů, která může být uložena přímo v HTML kódu nebo v samostatném souboru. Pomocí CSS můžeme měnit většinu vlastností od velikosti, stylu, barvy, okraje, přesné umístění na stránce a další.

Existují tři způsoby vložení kaskádových stylů do stránky:

- CSS v HTML tagu jako atribut
  Např.: Ukázkový text

- Pomocí stylopisu v hlavičce dokumentu:
  <style type="text/css">
  body { background-color: #fff; }
  table { padding: 0px; border-style: collapse; }
  </style>

- CSS uložené v externím souboru
  div.navigation
  {
  font-size: 15px;
  font-color: #00ff00
  }

  V HTML vypadá použití takto:
  text

2.2 Databázový systém

Hlavním úkolem databází je poskytnout uživatelům abstraktní pohled na data. Uživatel nepotřebuje znát fyzické rozložení a způsob uložení dat na disku, ale propojení a typ uložených dat. K přístupu, změnám a manipulaci záznamů slouží SŘBD (systém řízení báze dat), který také zajišťuje integritu a bezpečnost dat.

2.2.1 Databázové modely

Existují tři hlavní databázové modely a to hierarchický, síťový a relační.

2.2.1.1 Hierarchický model

Jedná se o nejstarší z uvedených modelů. Toto pojetí pochází z reálného uspořádání světa. Pro tento model je typická práce se stromy, kdy ve stromu jsou realizovány vztahy typu 1:N. Příkladem hierarchické databáze může být IMS od firmy IBM či M-system od firmy Micronetics.

2.2.1.2 Síťový model

Síťový model je variací hierarchického modelu databáze. V síťovém modelu je možné vyjadřovat vztahy 1:N a M:N. Nevýhodou tohoto modelu je náročná fyzická realizace a složitá aktualizace. Mezi výhody patří rychlost přístupu k záznamům.

2.2.1.3 Relační model

Relační databáze je považována za nejpropracovanější model databáze, který byl vyvinut doktorem E. F. Coddem v šedesátých letech minulého století.

Relační model definuje způsob, jakým je možné reprezentovat strukturu dat a dále ochranu a operace, které můžeme nad daty provádět. Relační databáze se skládá z řady tabulek, jejichž sloupce mohou být vázány na sloupce v jiné tabulce. Výhodou relačních databází jsou jednoduchá modifikace a propojování tabulek. Mezi nevýhody patří nízká efektivnost zpracování dotzazů (řada příkazů vyžaduje velké množství přístupu na disk, čímž se zpomaluje zpracování).

2.2.2 PostgreSQL

PostgreSQL je open-source databáze, která podporuje většinu ze standartu SQL 2003. Jedná se o relační databázi založené na POSTGRES vyvíjené na University of California at Berkley. POSTGRES je průkopníkem v mnoha funkcích, jenž jsou k dispozici v jiných komerčních databázích až později.

Přednostmi databáze PostgreSQL je podpora všech moderních operačních systémů (od verze 8.0.1 i win32) včetně OS/2 a Novell (je součástí většiny distribucí Linux). PostgreSQL je šířen pod GNU licencí umožňující vlastní úpravy a šíření binárního kódu, splňuje kritéria ACID testu. Z SQL podporuje pohledy, spouště, cizí klíče, JOIN tabulek, vnořené příkazy SELECT, příkazy CASE a NULLIF. Uživatelem definované funkce (UDF) mohou být psané v několika programovacích jazycích. UDF lze využít k návrhu konverzních a agregačních funkcí

a vlastních datových typů. Obsahuje kvalitní a přehlednou dokumentaci a český překlad FAQ (Frequently Asked Questions).

2.3 Počítačové sítě

Počítačová síť je systém sloužící ke komunikaci mezi počítači. Sítě mohou být stálé, nebo dočasné (pomocí modemů).

2.3.1 Linux

Linux je operační systém. Je to jeden z nejrozšířenějších operačních systémů, který je zdarma. Na rozdíl od operačních systémů jako jsou Windows, veškeré zdrojové kódy, které tvoří základ Linuxu, jsou dostupné veřejně a kdokoliv je může zdarma používat, měnit, vylepšovat
a znovu distribuovat.

Ze začátku byl Linux vyvíjen jen několika nadšenci. Poté byl podporován největšími počítačovými společnostmi jako jsou Sun Microsystems, IBM, Hewlett-Packard a Novel pro použití na serverech a stal se populárním také na osobních počítačích. Tento úspěch dosáhl Linux nejspíše díky nezávislosti, ceně, bezpečnosti a spolehlivosti.

Zakladatelem Linuxu je Linus Benedict Torvalds narozen 28. prosince 1969
v Helsinkách ve Finsku.

2.3.2 Traffic control

Ke kontrole rychlosti připojení slouží **tc** (traffic control). Pro použití traffic control musí být linuxové jádro na routru zkompilované s podporou classful qdisc HTB.

Ovlivnit tok dat můžeme pouze u odchozích dat, nemáme možnost omezovat příchozí data, což je dáno způsobem, jakým Internet funguje. Je to jako vaše (fyzická) poštovní schránka, nemůžete ovlivnit kolik pošty vám ostatní lidé posílají, ale můžete kontrolovat, kolik pošlete vy.

Základem Internetu je protokol TCP/IP, jehož některé funkce jsou využívány k omezení rychlosti. TCP/IP nezná kapacitu sítě mezi dvěma počítači, takže nejdříve začne posílat data pomalu a poté pokračuje rychleji a rychleji až do doby, kdy se začnou ztrácet pakety kvůli nedostatku šířky pásma, a proto rychlost zasílání dat sníží. Systém ovládání rychlosti je o něco složitější avšak vychází z předešlého popisu.

Traffic control, můžeme říci, zneužívá této vlastnosti TCP/IP a pokud potřebuje snížit rychlost u některého z počítačů, jednoduše začne zahazovat pakety a to v takové míře, až jsou data zasílána rychlostí, kterou požaduje. Avšak pozor, zahazování UDP nebo ICMP paketů

nevede ke snižování rychlosti zasílání paketů, tento způsob ovládání paketů je možné provádět jen na TCP paketech.

3 Realizace

3.1 Specifikace požadavků

Systém bychom měli rozdělit na tři nezávislé části - SQL databáze, grafické rozhraní pro ovládání celého systému a skripty, které slouží k nastavení routru (případně serverů). Celý systém by měl být založen na modulech, což umožňuje snadné rozšíření a přizpůsobení. Navíc si může kdokoliv naprogramovat nový či pozměnit stávající modul vlastním potřebám sítě.

Systém by měl také omezovat přístup uživatelů do některých částí tím, že umožní přiřadit přístupová práva k jednotlivým modulům. Všechny části systému by měli být zdarma, tzn. open- source operační systém a SQL databáze.

3.2 Analýza požadavků

Ze specifikace plyne potřeba tří částí:

- **SQL databáze** - jako databázi jsem si vybral PostgreSQL, protože ze všech databází dostupných zdarma poskytuje pokročilé funkce jako databázové triggery, funkce, stored procedures, foreign keys a další. Dále má některé datové typy, které se nenachází v jiných SQL databázích a to například *inet* pro ukládání IP adresy a *macaddr* pro ukládání fyzické adresy síťové karty (MAC adresy).
- **Grafické rozhraní** - slouží k ovládání celého systému, konfiguraci, přidávání a administraci uživatelů a spousty dalších vlastností. Grafické rozhraní je tvořeno ve skriptovacím jazyce PHP, kvůli jeho dobrým vlastnostem a rozšířenosti. Navíc server k interpretaci PHP může běžet na většině OS (např. zdarma dostupný server Apache).
- **Skripty** - jako operační systém pro router byl vybrán Linux, nabízí všechny potřebné programy a funkce, které jsou vyžadované u routru – routování, traffic control, DNS, DHCP. Navíc je také zdarma. Ke konfiguraci jsem nejdříve vybral Perl, kvalitní skriptovací jazyk s dobrou podporou v Linuxu, avšak praktičtější se nakonec ukázal skriptovací jazyk PHP. PHP interpret zabírá méně místa na disku, což je požadováno například na Wifi routerech, kde minimální verze operačního systému běží na flash disku a kapacita těchto zařízení je velmi malá.

3.3 Databáze

Samotný systém má pouze jednu tabulku *configuration*. Ta slouží k ukládání některých záznamů systému a modulů. Mezi systémové položky uložené v této tabulce patří název systému a jeho verze.

Tabulka *configuration*

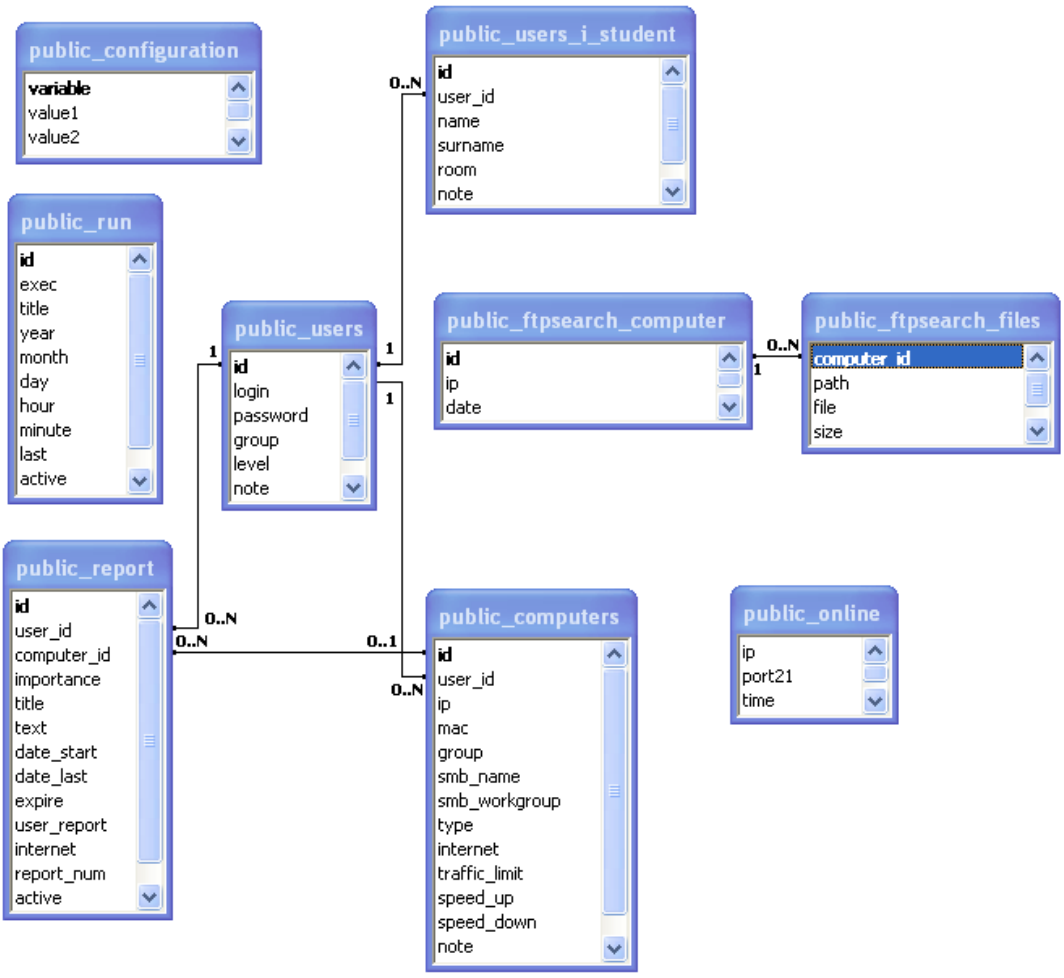
Obsahuje tyto sloupce:

- **variable** [varchar(255)] - primární klíč, unikátní identifikátor položky
- **value1** [varchar(255)] - první hodnota proměnné
- **value2** [varchar(255)] - druhá hodnota proměnné

Další tabulky v databázi jsou již tabulky jednotlivých modulů, seznam těchto tabulek závisí na nainstalovaných modulech v systému.

3.3.1 ER Diagram

ER Diagram je grafickým vyjádřením entit a vztahu mezi nimi. Jako entitu považujeme objekt či událost, o které chceme ukládat informace v databázi. Vlastnosti každé entity nazýváme atributy. Fyzickou reprezentací entity v databázi je relační tabulka.



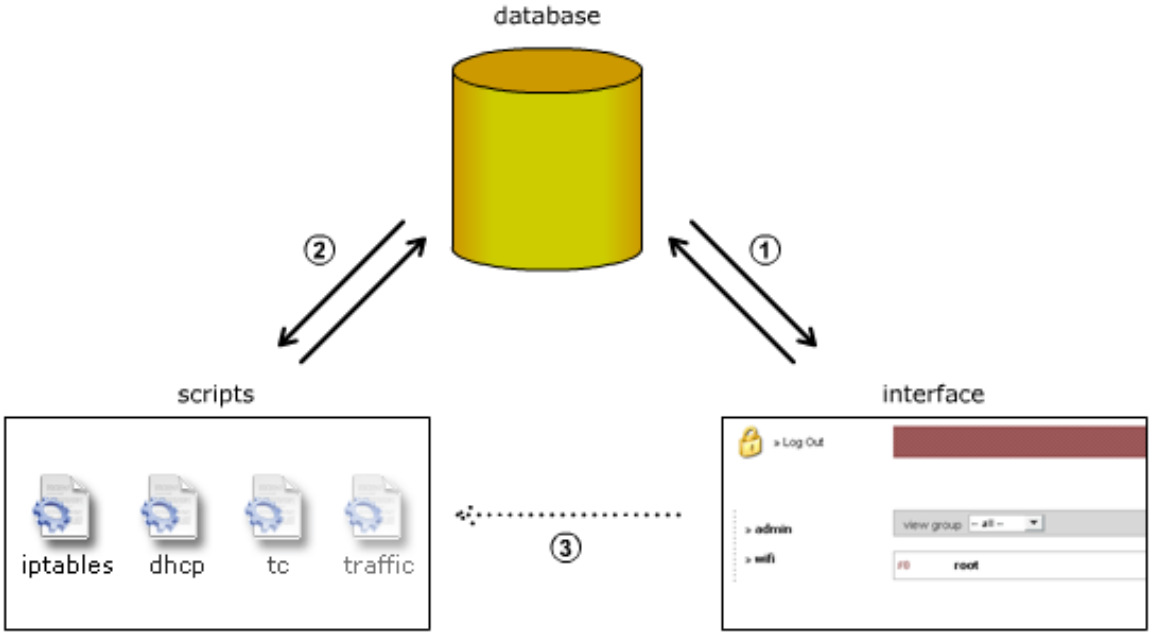
obr. 2. - ER Diagram (tabulky nainstalovaných modulů run, router, report, Admin a online)

3.4 Systém

Systém je rozdělen na tři rozdílné části. Za nejdůležitější bychom mohli považovat databázi PostgreSQL, kde jsou uloženy veškerá nastavení routru případně serverů, seznam uživatelů, jejich vlastnosti, práva, kontaktní údaje, připojené počítače a další.

Druhou částí je grafické rozhraní, pomocí nějž může administrátor jednoduše a přehledně měnit nastavení a nemusí tak znát jednotlivé nástroje operačního systému Linux.

Za třetí podstatnou část bychom mohli označit skripty, nahrané přímo na routru, které jsou propojené s databází, ze které čtou veškerá potřebná data a nastavují router.



Obr. 3. - spolupráce všech tří částí systému

Tento způsob rozdělení celého systému na tři části nám umožňuje nahradit každou z těchto tří částí jinou, tzn. například ovládací rozhraní v PHP můžeme nahradit jiným například v ASP.NET, které komunikuje s PostgreSQL. Nebo skripty v PHP pod Linux nahradit novými pod FreeBSD nebo Windows Server 2003, takže celá síť může běžet na stejném systému avšak pod jiným OS. Databáze je tedy tou hlavní částí celého systému, avšak i zde jsou prostředky pro nahrazení za jiný typ databáze. PHP rozhraní a PHP skripty využívají OOP (objektově orientované programování) a pro přístup k databázi slouží jeden globální objekt. V případě použití jiného typu databáze (např. MySQL) stačí do systému nahrát novou třídu pro přístup k této databázi odvozené od interface *database*.

3.4.1 Instalace systému

Instalace se skládá ze dvou kroků – 1) vytvoření databáze, tabulek a instalace modulů a 2) nahrání skriptů na router/server.

Vytvoření databáze, tabulek a instalaci modulů zajišťuje instalační rozhraní v PHP. Zde můžete zadat přihlašovací jméno a heslo pro přístup systému k databázi, případně odstranit starou databázi nebo přejmenovat a vytvořit databázi, systémové tabulky a tabulky pro jednotlivé moduly. Pokud chcete vybrat, které moduly mají být nainstalovány v průběhu

instalace systému, stačí přidat do složky */install/@plugins* instalační soubory (*id-název_pluginu**.ins.php) jednotlivých modulů.

Složka */install/system* by měla být po instalaci z bezpečnostních důvodů ručně odstraněna.

Instalační stránka se skládá ze tří částí, v hlavičce je seznam kroků s označením aktuálního kroku. Na konci stránky se také nachází seznam kroků, který umožňuje navrácení zpět v instalaci a přepínač pro zobrazování chyb při instalaci. Hlavní část se nachází uprostřed. Zde jsou zobrazeny informace o stavu instalace, formuláře pro uživatelské vstupy a další informace.

3.4.1.1 Kroky instalace

1) splash screen - zahájení instalace
2) připojení k databázi s právy administrátora
    a. IP adresa databázového serveru (localhost v případě, že systém běží na stejném počítači jako databáze)
    b. přihlašovací jméno uživatele s právy pro odstranění a vytvoření databáze
    c. heslo pro přístup k databázi
    d. název databáze systému
3) testování systému
    a. verze PHP (minimální požadovaná verze je 5, podpora objektově orientovaného programování (OOP))
    b. pokud databáze se stejným jménem již existuje, může uživatel zadat nové jméno databáze nebo existující databázi odstranit
4) vytvoření databáze, systémových tabulek a modulů. Nainstalovány jsou všechny moduly, jejichž instalační soubory se nachází ve složce /install/@plugins. Pokud dojde k chybě při instalaci některého z modulů, systém se nenainstaluje, databáze se odstraní a instalační soubor modulu, ve kterém nastala chyba, musí být odstraněn nebo zaktualizován
5) Zadání přihlašovacího jména a hesla pro přístup k databázi netysu (z bezpečnostních důvodů se doporučuje vytvořit nového databázového uživatele pro přístup pouze k databázi systému netys)
6) uložení konfiguračního souboru */_db.cfg.php*. V tomto souboru se nachází IP adresa SQL serveru, přihlašovací jméno a heslo a název databáze

3.4.2 Instalace a odstranění modulů

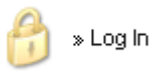
Na adrese *http://_adresa_serveru_/netys/install/plugins* můžete nainstalovat nebo odinstalovat jednotlivé moduly do nebo z již nainstalovaného a běžícího systému. Pro tuto akci je nutné mít ve složce */install/@plugins* instalační soubory jednotlivých modulů.

Pro přístup k instalačnímu rozhraní musí být uživatel přihlášen do systému jako root, tzn. level menší nebo roven 10.

3.5 Popis modulů

Vzhledem k požadavkům musí být systém přizpůsobitelný různým typům sítí a jejich specifickým potřebám. Jediným možným řešením je tedy systém založený z převážné části na modulech.

Samotný systém je jednoduchý a poskytuje modulům dva vytvořené objekty a to $DB pro práci s databází a $cfg pro rychlé čtení záznamů z tabulky *configuration*. Dále má na starost autorizaci uživatelů (ve spojení s modulem Admin) a načítání modulů.



uživatel není přihlášen

uživatel je přihlášen do systému

Moduly jsou načítány systémem v závislosti na hodnotě proměnné $q přenášené mezi relacemi:

1. systém testuje, zda proměnná $q existuje, pokud ne, není načten žádný z modulů
2. systém vyhledá daný modul v tabulce *configuration*, pokud modul není nelezen, tzn. že nebyl nainstalován, žádný modul není načten
3. načtení modulu do stránky (příkaz *include*)

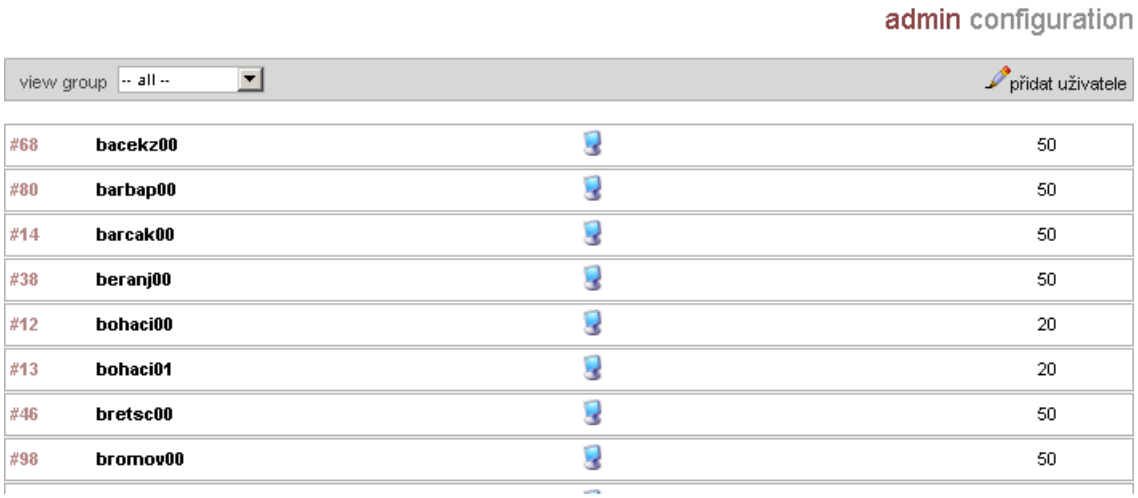
3.5.1 Admin

Základem každého systému je modul *admin*. Tento modul umožňuje přidávat uživatele, počítače a kontaktní údaje.

Na první stránce modulu admin je seznam všech uživatelů. Pro přehlednost si lze nechat zobrazit jen uživatele z některé ze skupin. V tomto seznamu je uvedeno ID uživatele, přihlašovací jméno (login), počet ikon počítače udává počet počítačů a nakonec práva.

Dále se zde nachází odkaz pro přidání nového uživatele, po kliknutí jste vyzván pro zadání přihlašovacího jména uživatele. Po zadání je vytvořen nový uživatel s daným přihlašovacím jménem, který nepatří do žádné skupiny a má práva 50.

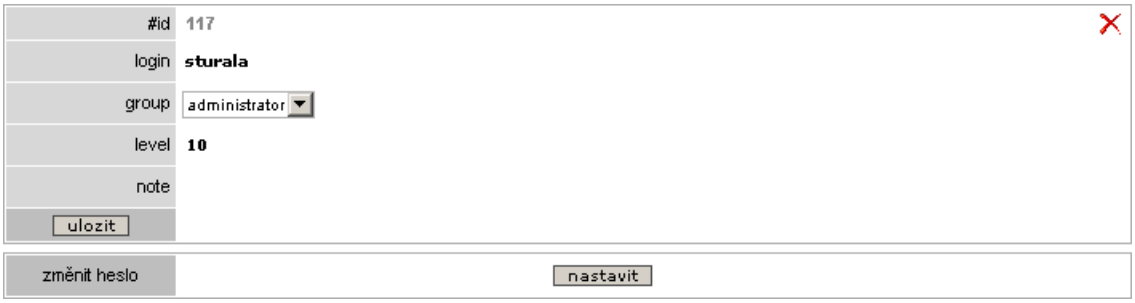
Po kliknutí na uživatele ze seznamu jsou zobrazeny údaje týkající se daného uživatele, kontakty a počítače. V pravém horním rohu se nachází drop-down menu pro přidání nového uživatele, přidání nového počítače danému uživateli nebo kontaktů.



obr. 4. - seznam všech uživatelů

Mezi povinné údaje každého uživatele patří jeho přihlašovací jméno, heslo a přístupová práva (level). Root, který má přístup do každého modulu, má práva 10 a musí být v každém systému alespoň jeden (systém nedovolí odstranit všechny uživatele root). Vyšší číslo znamená nižší práva v systému.

Odstranit uživatele je možné kliknutím na červený křížek v pravém horním rohu.



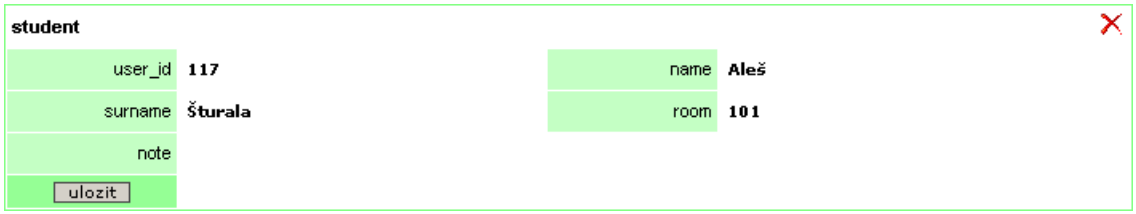
obr. 5. - minimální údaje povinné u každého uživatele

Dále v tomto modulu můžete přidávat kontakty k jednotlivým uživatelům. V různých systémech potřebuje administrátor ukládat různé kontakty o uživatelích. Například v městské

Wifi síti bude potřeba uložit jméno, příjmení, ulici, číslo popisné pro osoby nebo ICO, DIC a název pro firmu. Naopak v kolejní síti zase jméno, příjmení, číslo pokoje a studijní ID.

Systém umožňuje administrátorovi nadefinovat, které údaje chce o uživatelích ukládat. Stačí v databázi vytvořit tabulku a podle té se v grafickém rozhraní vygeneruje formulář pro zobrazení, editaci a odstranění záznamu. Systém pozná, které tabulky v databázi slouží k ukládání kontaktů o uživatelích podle názvu tabulky, přesněji podle prefixu. Prefix lze nastavit v konfiguraci, proměnná *admin_table_info_prefix*. Každá takováto tabulka musí povinně obsahovat sloupec *user_id*, který určuje ke kterému uživateli dané kontakty patří. Další sloupce jsou volitelné.

Odstranit kontakty lze kliknutím na červený křížek v pravém horním rohu.



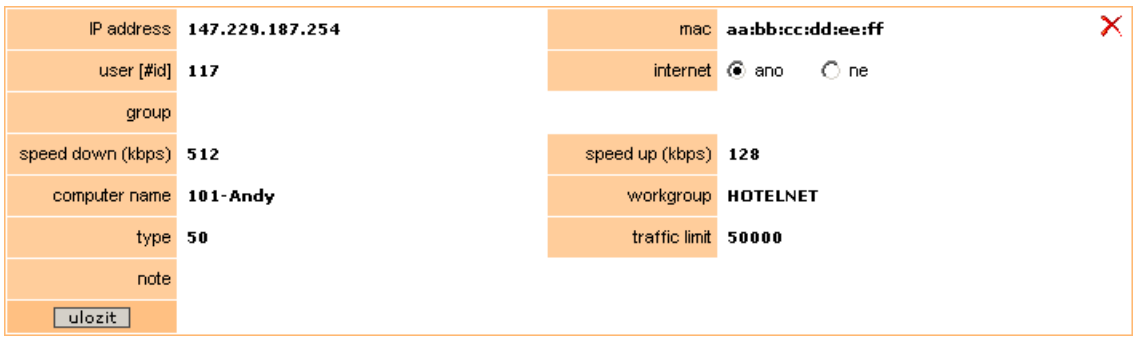
obr. 6. - dynamicky vygenerovaný formulář podle struktury tabulky v databázi

id	smallint
user_id (reference na tabulku users[id])	smallint
name	varchar(20)
surname	varchar(30)
room	smallint
note	varchar(255)

obr. 7. - struktura tabulky users_i_student, podle které byl vygenerován formulář

Ke každému uživateli je možné přidat neomezený počet počítačů. U každého počítače je možné nastavit IP adresu, fyzickou adresu síťové karty (MAC adresu), rychlost upload a download (*na* Internet a *z* Internetu) v kbps (kilobytes per second), dále typ počítače (server, router, osobní počítač), připojen k Internetu (ano/ne) a datový limit v MB (megabytes). Počítače mohou být seskupovány do skupin pomocí pole group. Pole user [#id], jak již název napovídá, udává vlastníka počítače, přesněji ID uživatele, kterému počítač patří. Změnou tohoto údaje na

ID jiného uživatele v systému dojde ke změně vlastníka počítače. Položky computer name (název počítače) a workgroup (pracovní skupina) jsou údaje, jakými se počítač identifikuje v síti a změnou těchto údajů dojde jen ke změně v databázi, ne na daném počítači. Dalším polem je note (poznámka), kde může být uvedena jakákoliv informace.



obr. 8. - počítač uživatele

Databáze

Tabulka *users*

V tabulce *users* jsou uloženy základní informace o všech uživatelích v systému, není závislá na jiných tabulkách.

Obsahuje tyto sloupce:

- **id** [smallint] - primární klíč k jednoznačné identifikaci uživatele
- **login** [varchar(255)] - přihlašovací jméno uživatele
- **password** [char(32)] - 32 bajtový MD5 hash. Heslo není v databázi uloženo v **textové** podobě ale jako MD5 hash z důvodu bezpečnosti, jedná se o jednosměrné šifrování.
- **group** [varchar(30)] - skupina, do které uživatel patří
- **level** [smallint] - přístupová práva
- **note** [varchar(255)] - dočasná informace, jakákoliv poznámka

Tabulka *computers*

Zde jsou uloženy data týkající se uživatelských počítačů. Data v tabulce závisí na tabulce *users*.

Obsahuje tyto sloupce:

- **id** [smallint] - primární klíč sloužící k jednoznačné identifikaci počítače
- **user**_id [smallint] - pomocí tohoto sloupce je tabulka *computers* propojena s tabulkou *users* a určuje vlastníka daného počítače v síti
- **ip** [inet] - IP adresa počítače v síti
- **mac** [macaddr] - fyzická adresa síťové karty počítače (také MAC adresa)
- **group** [varchar(255)] - skupina - počítače můžeme seskupovat do skupin
- **smb_name** [varchar(255)] - název počítače (samba) - změnou tohoto údaje v systému nezměníte název nastavený na daném počítači
- **smb_workgroup** [varchar(255)] - pracovní skupina (workgroup) - změnou tohoto údaje v systému nezměníte pracovní skupinu nastavenou na daném počítači
- **type** [smallint] - funkce počítače v síti - osobní počítač, router, server
- **internet** [boolean] - udává zda je daný počítač připojen k Internetu nebo není připojen
- **traffic_limit** [integer] - datový limit v MB/měsíc
- **speed_up** [smallint] - rychlost připojení k Internetu v kbps (kilobits per second) - upload
- **speed_down** [smallint] - rychlost připojení k Internetu v kbps (kilobits per second) - download
- **note** [varchar(255)] - dočasná informace, jakákoliv poznámka

3.5.2 Report

Tento modul slouží k omezení připojení k Internetu na určitou dobu a s uvedeným důvodem. Každý report má určeno, zda má být zakázán přístup k Internetu, název reportu a případně text s podrobným popisem. Dále počet výskytů, doba prvního nahlášení, poslední nahlášení a doba platnosti (např. doba platnosti 5 hodin znamená, že report vyprší 5 hodin po posledním nahlášení, prázdná hodnota znamená na neurčito). Také zda se jedná o report počítače nebo uživatele (u uživatele v případě, že je zakázáno připojení k Internetu je Internet odpojen od všech počítačů daného uživatele).

Každý report patří do jedné ze tří skupin podle závažnosti od nejnižší po nejvyšší - oznámení (announcement), výstraha (warning) a varování (alert). Do varování patří reporty s nejvyšší závažností, do výstrahy se střední závažností a do oznámení pouze informační

reporty. Tento modul využívají především skripty na routru, které slouží k dodržení pravidel sítě nebo můžete vložit report ručně pomocí formuláře.

Příklad: Máme síť, kde jedním z pravidel je dodržení jednotného názvu pracovní skupiny (workgroup) počítačů v síti. Skript běžící na routru jednou za hodinu testuje, zda je pracovní skupina počítačů v síti nastavena na správnou hodnotu. V případě, že tomu tak u některého z počítačů není, založí nový report pro daný počítač (pokud již report existuje, zvýší počet výskytů o 1 a zaktualizuje datum posledního výskytu) v kategorii *warning*, nastaví hodnotu prvního výskytu a posledního výskytu na aktuální datum a čas, odpojí Internet a dobu platnosti daného reportu nastaví na 1 hodinu.

Databáze

Tabulka *report*

V tabulce *report* jsou uloženy všechny záznamy a informace modulu *report*.

Obsahuje tyto sloupce:

- **id** [smallint] - primární klíč k jednoznačné identifikaci reportu (autoincrement)
- **user_id** [smallint] - uživatel, kterému daný report patří. Pomocí tohoto sloupce je tabulka *report* propojena s tabulkou *users*.
- **computer_id** [smallint] - Pomocí tohoto sloupce je tabulka *report* propojena s tabulkou *computers*. Pokud je hodnota null, report je vystaven na uživatele, tzn. na všechny jeho počítače.
- **importance** [smallint] - kategorie, do které report patří. Hodnota 10 znamená announcement, 20 je warning a 30 je alert.
- **title** [varchar(255)] - název reportu
- **text** [text] - text reportu, případně podrobný popis
- **date_start** [timestamp] - datum a čas prvního nahlášení reportu
- **date_last** [timestamp] - datum a čas posledního nahlášení reportu
- **expire** [smallint] - udává, kdy daný report vyprší. Hodnota 0 znamená, že report nevyprší nikdy, jiná hodnota udává počet hodin, který musí uplynout od posledního nahlášení
- **user_report** [boolean] - udává, zda daný report může být zobrazen uživateli nebo pouze administrátorovi

- **internet** [boolean] - udává, zda má být uživatel odpojen od Internetu, pokud ano, je odpojen do doby, než daný report vyprší
- **report_num** [smallint] - počet nahlášení reportu
- **active** [boolean] - udává, zda daný report je aktivní nebo již vypršel

3.5.3 FTP search

Modul FTP search slouží k vyhledávání souborů na síti, přesněji jde o statické vyhledávání dat na všech FTP serverech připojených do sítě. FTP search neprohledává data na všech FTP serverech v reálném čase, kdy uživatel zadá požadavek na vyhledání řetězce, ale FTP servery prohledává v určitých časových intervalech a ukládá strukturu a soubory do databáze.

V konfiguraci je možné nastavit rozmezí IP adres, ve kterém se mají dané FTP servery nacházet (*ftpsearch_ip_range*). Skript nahraný na routru prochází každou IP adresu v daném rozmezí a pokud se úspěšně přihlásí k FTP serveru, prochází rekurzivně každou složku a ukládá strukturu, velikost souborů a názvy souborů do databáze. Rekurzivní procházení je omezeno na hodnotu uvedenou v konfigurační proměnné *ftpsearch_screw*. Například hodnota 5 znamená, že prohledávání proběhne maxilmálně do páté podsložky. Přihlašovací jméno a heslo, pod kterým se vyhledávač pokouší přihlásit na FTP server, jsou nastaveny v konfigurační proměné *ftpsearch_login*.

Prohledávány jsou jen servery, které ještě nejsou v databázi nebo jsou v databázi údaje starší více než 2 dny (lze ovlivnit změnou hodnoty konfigurační proměné *ftpsearch_update*). Data v databázi starší více než pět dnů jsou automaticky odstraněny (konfigurační proměnná *ftpsearch_delete*).

Vzhledem k obrovskému počtu souborů uložených na FTP serverech systém ukládá do databáze jen některé známé typy souborů, velikost databáze je proto menší. Do databáze je tedy ukládána struktura (složky) a soubory s těmito příponami:

bmp, rle, dib, psd, pdd, gif, eps, raw, jpg, jpeg, jpe, pcx, pdf, pdp, pxr, png, sct, pct, pict, tga, vda, icb, vst, tif, tiff, exe, bat, com, txt, doc, sxw, sub, srt, chm, xls, zip, rar, arj, scr, swf, fla, html, htm, php, php3, php4, php5, phtml, asp, aspx, cs, vb, sql, sdf, mdf, ini, wav, mp2, mp3, mp4, mid, avi, asx, wmv, qt, pl, asm, pas, xls

Vyhledávání souborů probíhá přes webové rozhraní. Po zadání vyhledávaného řetězce jsou vypsány složky a soubory, ve kterých se daný řetězec vyskytuje. Ve výpisu je IP adresa FTP serveru, úplná cesta ke složce nebo souboru, velikost souboru a datum a čas aktualizace daného FTP serveru. Modul FTP search spolupracuje s modulem online a v případě, že modul

online je nainstalován v systému, jsou zobrazené výsledky rozdělené do dvou částí podle toho, zda je daný FTP server právě přístupný nebo je vypnutý.

Databáze

Tabulka *ftpsearch_computer*

V tabulce *ftpsearch_computer* jsou uloženy všechny FTP servery, jejichž struktura a soubory byly uloženy do databáze.

Obsahuje tyto sloupce:

- **id** [smallint] - primární klíč k jednoznačné identifikaci počítače (autoincrement)
- **ip** [inet] - IP adresa počítače v síti, musí být unikátní
- **date** [date] - datum aktualizace daného FTP serveru

Tabulka *ftpsearch_files*

V tabulce ftpsearch_files je uložena celá struktura složek a souborů na daném FTP serveru a případně velikost souborů.

Obsahuje tyto sloupce:

- **computer_id** [smallint] - primární klíč k jednoznačné identifikaci počítače, pomocí tohoto sloupce je tabulka *ftpsearch_files* propojena s tabulkou *ftpsearch_computer*
- **path** [varchar(255)] - úplná cesta ke složce nebo souboru
- **file** [varchar(255)] - název složky nebo souboru
- **size** [integer] - pokud se jedná o soubor, je v tomto sloupci uložena velikost souboru v KB (kilobytes), pokud se jedná o složku, je v tomto sloupci uložena hodnota -1. Naopak podle této hodnoty se při vyhledávání rozlišuje, zda se jedná o složku (size = -1) nebo soubor (size >= 0)

3.5.4 online

Zobrazí seznam počítačů, které jsou připojené k síti. Volitelně může zobrazit vlastníka počítače.

Konfigurační proměnné:

online_ip_range - rozsah testovaných IP adres (ve formátu pro program nmap)

online_ftp_default_login - přihlašovací jméno a heslo na FTP server

online_show_user - pokud je hodnota 1 tak ve výpisu je zobrazen vlastník počítače, v opačném případě vypsán není

3.6 Skripty

3.6.1 traffic control

tc.run.php

Jedná se o jeden z nejdůležitějších skriptů, který slouží k nastavení rychlosti připojení jednotlivých počítačů k Internetu.

Pro bezproblémové fungování skriptu na omezování rychlosti musí Linux (router) běžet na jádře s podporou HTB, tedy 2.4.20 a vyšší a se sadou nástrojů *iproute* nebo opatchovaném starším jádře.

Vyžaduje moduly **router** a **admin**.

- **router** - vlastnosti *upload* a *download* - rychlost připojení sítě k Internetu (odchozí/příchozí), *lan rozhraní* a *Internet rozhraní* a *ceil*
- **admin** - nastaví rychlost připojení počítače k Internetu podle rychlosti uvedené v *speed up* a *speed down* (v kbps).

Rychlost uvedená u počítače je však jen relativní. Většinou nastane situace, že součet rychlostí *up* nebo *down* všech počítačů bude vyšší než rychlost připojení celé sítě k Internetu *up* nebo *down* nebo naopak bude nižší. Řešením je výpočet koeficientu (ratio).

Příklad pro download:

koeficient = rychlost_pripojeni_site_down/soucet(rychlost_pripojeni_kazdeho_pocitace_down)

koeficient < 1: tzn. součet rychlostí připojení všech počítačů je *vyšší* než je rychlost připojení celé sítě

důsledek: rychlost připojení počítače bude *nižší*

koeficient > 1: tzn. součet rychlostí připojení všech počítačů je *nižší* než je rychlost připojení celé sítě

důsledek: rychlost připojení počítače bude *vyšší*

garantovana_rychlost_pocitace = zadana_rychlost_pocitace * koeficient

Takto tedy dojde k rozložení rychlosti mezi všechny počítače v poměru jejich přiřazené rychlosti. Jedná se o tzv. garantovanou rychlost, tzn. že vždy máme zaručeno, že rychlost připojení počítače nebude menší než garantovaná rychlost připojení. Avšak musíme si uvědomit, že nikdy nedojde k situaci, kdy všechny počítače na síti využijí 100% svého přiřazeného pásma. Můžeme říci, že v jeden okamžik bude svého přiřazeného pásma využívat asi 5-10% uživatelů, můžeme však rozdělit i toto nevyužité pásmo mezi připojené počítače. Způsob rozdělení nevyužité rychlosti není v poměru přiřazené rychlosti, jak bychom očekávali, ale každému spojení je přiřazená stejná rychlost. Bohužel různé programy mají různý počet spojení a například počítač s programem StrongDC nebo BitCommet mohou mít vytvořeno 20-30 spojení pro stahovaní jednoho souboru a tak rychlost přiřazená je 20-30x rychlejší než rychlost počítače, kde stahování probíhá přes HTTP (1 spojení). Řešením je nastavení horní hranice rychlosti. V systému netys je to určení násobku reálné rychlosti (položka ceil).

Příklad: Máme v síti dva počítače A a B. Síť je připojena rychlostí 1024kbps k Internetu. Rychlost sítě rozdělíme na 256kbps pro upload a 768kbps pro download. Počítači A zadáme rychlost 64kbps upload a 128kbps download a počítači B 128kbps upload a 256kbps download. Vidíme, že přiřazená rychlost pro upload obou počítačů dohromady je jen 192kbps avšak celé sítě je 256kbps, takže 64kbps je nevyužito. Systém ale automaticky přidělí celé pásmo počítačům v poměru jejich zadané rychlosti.

koeficient_upload = 256 / (64 + 128) = 1.33
koeficient_download = 768 / (128 + 256) = 2

Počítač A:
garantovaná_rychlost (upload) = 64 * 1.33 = 85.12kbps
garantovaná_rychlost (download) = 128 * 2 = 256kbps

Počítač B:
garantovaná_rychlost (upload) = 128 * 1.33 = 170.24kbps
garantovaná_rychlost (download) = 256 * 2 = 512kbps

Vidíme, že je opravdu využito celé šířky připojení sítě k Internetu (upload 85.12kbps + 170.24kbps = 256kbps a download 256kbps + 512kbps = 768kbps).

Hodnota ceil v této síti by měla být 2, tzn. že rychlost připojení počítače může být maximálně dvojnásobek garantované rychlosti. Ve velkých sítích je nastavení hodnoty ceil individuální.

Příklad: Síť je připojena k Internetu 45mbps (15.000kbps upload a 30.000kbps download). Největší zátěž sítě je mezi 17:00-21:00 hodin a zde počítáme asi s 10%-20% uživateli na síti, tzn. že 80%-90% garantované rychlosti je nevyužito. Hodnota ceil pro tuto síť by měla být asi 8, tzn. že rychlost připojení počítačů může být maximálně osminásobek jejich garantované rychlosti.

Nastavení vysoké hodnoty ceil může zapříčinit přidělení velké rychlosti počítačů s vysokým počtem spojení a malé rychlosti ostatním počítačům. Naopak nastavením nízké hodnoty ceil může dojít k nevyužití celého pásma připojení sítě.

tc.run.php -help

```
-help               : this help
-test               : only prints table of speeds for each
                      computer, does not set router
--ceil [value]      : multiple of speed (maximal computer speed up
                      and down)
                      if user's speed is 64kbps and ceil is 4 =>
                      maximum speed is 64*4=256kbps
--ceil-up [value]   : ceil for upload
--ceil-down [value] : ceil for download
```

3.6.2 FTP search

ftpsearch.run.php

Tento skript slouží k prohledávání FTP serverů v určeném rozmezí a ukládání struktury a souborů do databáze.

Vyžaduje modul FTP search.

3.6.3 online

online.run.php

Tento skript slouží k zobrazení připojených počítačů k síti a testuje zda je na daném počítači nainstalován a přístupný FTP server.

Vyžaduje modul online, který zobrazí počítače připojené k síti. Pokud je nainstalován také modul admin, zobrazí přihlašovací jméno uživatele, kterému daný počítač patří.

3.6.4 iptables

iptables.run.php

Slouží k připojení počítačů k Internetu. Nastavuje *iptables* pro přístup počítačů k Internetu a zároveň nastaví iptables tak, aby každý paket byl označen ID daného počítače (využití například při traffic control).

Vyžaduje modul Admin.

3.7 Vývoj modulů

Každý modul je obyčejný php soubor načtený systémem. Modul má k dispozici dva inicializované objekty. Objekt $DB slouží pro přístup k databázi a objekt $cfg pro čtení hodnot konfigurace.

Metody objektu $DB:

$DB->query(string $sql)

- metoda pro zadávání a vykonání příkazu v databázi. Zadávání příkazů může být v chráněném nebo nechráněném režimu. Chráněný režim nahrazuje všechny výskyty některých znaků.

z	*na*
"	!
<	«
>	»

  Jedná se o znaky využívané při SQL injections a XSS útocích. Pokud potřebujete použít některý z těchto znaků v SQL příkazu, metoda *security_off* chráněný režim vypíná, naopak metoda *security_on* chráněný režim zapíná
- parametr je sql příkaz
- vrací *resource* sql dotazu

$DB->fetch_object(resource $qu)

- ukládá přečtená data z databáze do objektu

- parametr typu resource sql dotazu
- vrací objekt

$DB->fetch_array(resource $qu)

- ukládá přečtená data do pole
- parametr typu resource sql dotazu
- vrací pole

$DB->affected_rows(resource $qu)

- parametr typu resource sql dotazu
- vrací počet řádků, kolik ovlivnil (změnil, odstranil, přidal) sql dotaz

$DB->num_rows(resource $qu)

- parametr typu resource sql dotazu
- vrací počet řádků, které vrací sql dotaz select

$DB->free_result(resource $qu)

- uvolnění dat z paměti databáze, vhodné v případě sql dotazů select, které vrací velký počet záznamů
- parametr typu resource sql dotazu

$DB->get_tables(string $table = '%')

- parametr typu řetězec, který udává název tabulky (znak '%' v názvu tabulky udává jakýkoliv řetězec místo znaku '%')
- vrací pole všech nalezených tabulek daného názvu

$DB->get_columns(string $table)

- parametr je název tabulky
- vrací pole všech sloupců dané tabulky (['column'] = název; ['type'] = datový typ)

$DB->last_error()

- vrací poslední error

$DB->close()

- uzavírá spojení s databází

$DB->drop_db(string $db_name)

- odstranění databáze
- parametr je název databáze

$DB->get_db(string $dbname = '%')

- parametr typu řetězec, který udává název databáze (znak '%' v názvu tabulky udává jakýkoliv řetězec místo znaku '%')
- vrací pole všech nalezených databází daného názvu

$DB->security_on()

- zapíná chráněné sql dotazy

$DB->security_off()

- vypíná chráněné sql dotazy

Metody objektu $cfg:

$cfg->get(string $var)

- načítá hodnoty všech proměnných s názvem $var
- parametr udává název proměnné (znak '%' v názvu proměnné udává jakýkoliv řetězec místo znaku '%')
- vrací pole všech proměnných s názvem $var

  $return['nazev_promenne'][0] - 1. hodnota proměnné
  $return['nazev_promenne'][1] - 2. hodnota proměnné

Dále jsou v systému proměnné identifikující uživatele:

$_SESSION['login_id'] - ID uživatele přihlášeného do systému, pokud uživatel není přihlášen, tato proměnná není inicializována

$_SESSION['login_user'] - přihlašovací jméno uživatele přihlášeného do systému, pokud uživatel není přihlášen, tato proměnná není inicializována

$_SESSION['login_level'] - práva uživatele, tato proměnná existuje vždy. Pokud uživatel není přihlášen, práva jsou nastaveny na hodnotu 100

4 Závěr

Cílem této bakalářské práce bylo zhodnotit metody vytváření informačních systémů v prostředí Internetu, seznámit se podrobněji s problematikou sítí, Linuxu a informačních systémů a na základě takto získaných znalostí realizovat univerzální systém pro tuto oblast s použitím skriptovacích jazyků PHP a JavaScript.

Kromě těchto jazyků jsem použil k implementaci také značkovací jazyk HTML a databázový server PostgreSQL, který je volně dostupný a plně postačuje pro potřeby ukládání dat malé počítačové sítě.

Důraz při tvorbě systému jsem kladl na funkčnost, přehledné uživatelské rozhraní a jednoduchou obsluhu, aby ovládání celého systému bylo co nejpříjemnější pro administrátora, případně uživatele. Existují také jakási nepsaná pravidla pro rozmístění a vzhled ovládacích prvků, kterých jsem se snažil držet. Uživatel se tak nemusí zdlouhavě zabývat ovládáním systému.

Při tvorbě jsem si rozšířil znalosti skriptovacích jazyků, především PHP, databázové komunikace a znalosti operačního systému Linux a některých jeho síťových nástrojů. Zdokonalil jsem se velmi i v oblasti celkového procesu vývoje dynamických Internetových aplikací.

Možnosti dalšího pokračování vidím v rozšíření počtu modulů, podpoře dalších databázových systémů nebo jiných operačních systémů (například FreeBSD) a lokalizaci systému.

5 Použité zdroje

[1] internetové stránky: http://www.wikipedia.org

[2] internetové stránky: http://www.root.cz

[3] internetové stránky: http://lartc.org