VŠB – TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
KATEDRA INFORMATIKY

PRAKTICKÉ NASAZENÍ DOKUWIKI
PRACTICAL IMPLEMENTATION OF DOKUWIKI

2012 Jaroslav Křibík

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Jaroslav Křibík**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Praktické nasazení DokuWiki
Practical Implementation of DokuWiki

Zásady pro vypracování:

Cílem bakalářské práce je návrh a realizace DokuWiki pro tým laboratoře IPtelefonie včetně autentizace uživatelů přistupujících do DokuWiki.

1. Publikování v Internetu - HTML, XHTML, XML, PHP, JAVA.
2. Ověření identity uživatele - LDAP, Radius, Shibboleth.
3. Návrh DokuWiki pro laboratoř IP telefonie.
4. Autentizace přístupu.
5. Praktická realizace navržené DokuWiki.

Seznam doporučené odborné literatury:

KOSEK, J. *XML pro každého*. Praha: Grada, 2000. ISBN 80-7169-860-1.
GROPL, T. *HTML, CSS a JavaScript*. Praha: BEN, 2002. ISBN 80-7300-099-7.
KOSEK, J. *PHP tvorba interaktivních internetových aplikací*. Praha: Grada, 1998. ISBN 80-7169-737-1.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Miroslav Vozňák, Ph.D.**

Datum zadání: 19.11.2010

Datum odevzdání: 04.05.2012

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

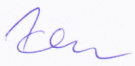
prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal."

V Ostravě: 4. 5. 2012

Jaroslav Křibík



Poděkování

Děkuji vedoucímu bakalářské práce doc. Ing. Miroslavu Vozňákovi, Ph.D. za podporu a nápady. Dále bych rád poděkoval Ing. Janu Rozhonovi za asistenci a pomoc při nasazování systému na katedrální hardware a Ing. Martinu Lasoňovi z útvaru CIT-IS za informace k VŠB-TU LDAP a zřízení testovacích účtů.

Abstrakt

Smyslem této bakalářské práce je nasadit systém DokuWiki pro potřeby vedení studentských semestrálních projektů na Katedře telekomunikační techniky. První část práce se věnuje obecné teorii publikování na internetu a s tím spojených webových technologií s navazující problematikou ověření identity a autentizace přístupu k systémům. V druhé části se pak práce zabývá samotným řešením nasazení systému v prostředí vnitřní sítě VŠB-TU. V závěru jsou pak shrnuty poznatky a doporučení.

Klíčová slova
Wiki, DokuWiki, LDAP, autentizace, webové technologie

Abstract

The purpose of this bachelor degree thesis is application of the system DokuWiki for the needs of management and administration of student semestral projects at the Department of telecommunications. The first part of the work deals with general theory of publishing on the internet and with web technologies related to it, with connected issues of identity verification and authentication of access to the systems. The second part deals with the solution of the system implementation in the environment of VŠB-TU internal network. Conclusions summarize the findings and recommendations.

Keywords
Wiki, DokuWiki, LDAP, authentication, web technologies

Seznam použitých zkratek a symbolů

ACL - Access control list
AD - Active Directory
AJAX - Asynchronous JavaScript and XML
API - Application Programming Interface (rozhraní pro programování aplikací)
CAPTCHA - Completely Automated Public Turing test to tell Computers and Humans Apart
CSS - Cascading Style Sheets
DHTML - Dynamic HyperText Markup Language (dynamický značkovací jazyk pro hypertext)
DIT - Directory Information Tree
DN - Distinguish Name
DSML - Directory Service Markup Language
DTD - Dokument Type Definition
ECMA - European Computer Manufacturers Association
GSSAPI - Generic Security Services Application Program Interface
GUI - Graphical User Interface (uživatelské rozhraní)
HTML - HyperText Markup Language (značkovací jazyk pro hypertext)
IEEE - Institute of Electrical and Electronics Engineers (Institut pro elektrotechnické a elektronické inženýrství)
ISO - International Organization for Standardization
ISP - Internet Service Providers (pokytovatel připojení k internetu)
JRE - Java Runtime Environment
JVM - Java Virtual Machine
LDAP - Lightweight Directory Access Protocol
LDIF - LDAP Data Interchange Format
OID - Object Identifier
PDA - Personal Digital Assistant (osobní digitální pomocník)
PHP - Hypertext Preprocessor
RADIUS - Remote Authentication Dial In User Service (uživatelská vytáčená služba pro vzdálenou autentizaci)
RDN - Relative Distinguished Name
SASL - Simple Authentication and Security Layer
SLP - Service Location Protocol
SOAP - Simple Object Access Protocol
SPML - Service Provisioning Markup Language
SQL - Structured Query Language
SSO - Single sign-on (systém jednotného přihlášení)
W3C - World Wide Web Consortium
WWW - World Wide Web (zkráceně web)
XAML - Extensible Application Markup Language
XED - XML Enabled Directory
XHTML - eXtensible HyperText Markup Language (rozšiřitelný hypertextový značkovací jazyk)
XML - Extensible Markup Language (rozšiřitelný značkovací jazyk)

Obsah

1. Úvod

Problematika vedení projektové dokumentace se v dnešní době převážně spoléhá na elektronické systémy. Základním předpokladem funkčnosti takového systému je uživatelská přívětivost, přehlednost a flexibilita. Pro potřeby Katedry telekomunikační techniky byl jako systém, který by mohl pomoci pedagogům i studentům s vedením dokumentace, navržen open source projekt DokuWiki.

Cílem této bakalářské práce je návrh a implementace systému pro potřeby vedení dokumentace studentských semestrálních projektů. Práce se v druhé kapitole věnuje popisu technologií dnes běžně používaných pro publikování na Internetu. Na tuto problematiku pak navazuje kapitola třetí, která se zaměřuje na otázku ověření identity v prostředí rozsáhlých sítí a popisuje základní pojmy, principy a technologie. Čtvrtá kapitola pak shrnuje poznatky, návrhy a řešení spojené s nasazením systému v prostředí sítě VŠB-TU. Problematice praktického řešení uživatelské autentizace a autorizace v navrhovaném projektu se věnuje kapitola pátá. Výsledný technický popis nasazení systému a zabezpečení jeho fungování je popsáno v šesté kapitole. V kapitole sedm jsou shrnuty poznatky a doporučení vyplývající z vlastní implementace systému.

2. PUBLIKOVÁNÍ V INTERNETU - HTML, XHTML, XML, PHP, JAVA

Dnešní Internet je globálním médiem zasahujícím do všech aspektů života moderního člověka. V rozvinutých společnostech se již stěžejní část veškeré společenské komunikace přesunula do prostředí tohoto elektronického média a úspěšně vytlačuje ostatní metody výměny informací. Informace tvoří po celou historii lidstva základní stavební kámen mezilidské komunikace a kvalitní informace je pak nutnou podmínkou úspěšného rozhodování jednotlivců i celých skupin. Internet a s ním spojené technologie přinesly homogenní médium, kde je informace distribuována a uchovávána v mnoha podobách, kdy záleží jen na výběru vhodné prezentační formy. Jestliže je Internet globálním systémem, který vzájemně propojuje jednotlivé počítačové sítě, pak web je aplikací propojující pomoci odkazů dokumenty, které jsou na něm publikovány.

Web je obecně vžité označení pro WWW (World Wide Web). Jedná se o vzájemně propojené hypertextové dokumenty, mezi kterými se v prostředí webového prohlížeče uživatel pohybuje pomocí odkazů (linků, hyperlinků). Základním dokumentem webu je webová stránka, která umožňuje logické a stylistické uspořádaní publikovaných informací skrze hypertext.

ZÁKLADNÍ ROZDĚLENÍ WEBU

- Statický web – informace jsou ve stejném tvaru publikovány a zároveň prezentovány, tedy forma je předem dána.
- Dynamický web – forma publikovaných informací je generována na základě požadavku, a to jak na straně klienta, tak na straně serveru.

HYPERTEXT

Je označení pro nelineární text, jehož části jsou propojeny za pomoci odkazu. Odkazy však nemusejí vést jen k informacím ve formě textu. Pod označením hypermédia zde patří i další objekty ve formě elektronických dat (obrázky, audio, video, programy, atd.).

TVORBA STANDARDIZOVANÝCH STRÁNEK

Tvorba standardizovaných stránek zajišťuje vytváření stránek na základě pravidel, díky kterým je pak zajištěna jednoznačnost významu a správnost vykreslování stránek. Význam vytváření webových stránek dle platných standardů má obrovský význam. Pomocí standardizace lze dosáhnout stejného vzhledu a chování stránek ve všech prohlížečích na jakémkoliv zařízení, ať už se jedná o klasické PC, chytrý mobilní telefon nebo moderní televizi. Tvorbou podle standardu rovněž zjednodušíme pozdější revizi kódu.

V elektronické podobě se jedná o přesnou definici vzhledu dokumentu tak, aby bylo jeho zobrazení (renderování) shodné napříč všemi systémy, které s ním, mohou pracovat. Značky

(markups) jsou součástí dokumentu a během procesu zobrazení jsou interpretovány na základě jejich významu. Samy značky však při výsledném zobrazení vidět nejsou a jsou pouze součástí zdrojového kódu dokumentu.

2.1. XML

ZNAČKOVACÍ JAZYK/ZNAČKOVÁNÍ/TAGY

Značkování je v běžné tiskové technologii proces vyznačení formy a typu pro potřeby sazeče (velikost písma, typ písma, zarovnání, atd.). Ten na základě těchto informací, které nejsou ve výsledném textu vidět, definuje výstupní typografický vzhled dokumentu, tak jak bude prezentován.

XML (eXtensible Markup Language) je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Je zjednodušenou podobou staršího jazyka SGML. Pravidla tohoto jazyka jsou tak volná, že umožňují jeho využití pro téměř jakékoliv použití – od prostého ukládání strukturovaných dat, až po zápis určitého kódu, což umožnilo vzniknout jazyům, jako jsou XHTML nebo XAML [1].

2.2. HTML/XHTML

Značkovací jazyk pro hypertext HTML (Hyper Text Markup Language) v současné době v běžném prostředí ve verzi 4.01 a nově se vyvíjející verzi HTML5 [2].

XHTML (eXtensible Hypertext Markup Language) měl být původně náhradou HTML, který byl původně ve verzi 4.01 ukončen. V roce 2007 však došlo znovu k nastartování vývoje HTML a XHTML ve verzi 5. V mezidobí mezi verzemi, než začal vývoj verze 5, probíhal vývoj XHTML 2.0, ale na základě rozjezdu vývoje verze 5, byl v roce 2009 ukončen.

Zásadní rozdíl mezi HTML a XHTML je v aplikaci pravidel jazyka XML na již existující standard HTML. Pravidla zápisu syntaxe odpovídají pravidlům jazyka XML, obsahovou stránku věci kontrolují DTD vytvořená W3C podle standardů. Zpřísnění syntaxe společně s možností nadefinovat si vlastní značky a skutečnost, že všechny párové značky jsou přísně párové a nepárové značky jsou vždy ukončeny lomítkem, zabezpečuje vysokou konzistenci obsahu a spolehlivou interpretaci bez chyb.

KONSORCIUM W3

Konsorcium W3 (plným jménem World Wide Web Consorcium, zkráceně W3C) je organizace zabývající se správou všech standardů týkajících se tvorby webových stránek. Má na starosti standardy jako je HTML/XHTML, CSS, XML, SOAP nebo samotný HTTP protokol. Konsorcium se skládá z mnoha organizací, které se zavázaly tyto standardy používat a nadále je rozšiřovat. Předsedou tohoto konsorcia je sir Tim Berners-Lee, zakladatel konsorcia a autor specifikací URL, HTTP a HTML [2].

VALIDACE STRÁNKY

Validace stránky ověřuje správnost kódu vzhledem k platným standardům. Kvůli nestandardnímu zápisu kódu může dojít k chybě vykreslování nebo odlišné interpretaci různými prohlížeči (k tomu může ovšem dojít i v případně validní stránky kvůli špatně navrženému prohlížeči jako takovému). Validní stránka se zobrazuje ve všech prohlížečích identicky. Validní kód má také kladný dopad i na rychlost načítání stránky. Validaci je možné provést buď pomocí nástrojů pro tvorbu webových stránek, nebo pomocí online služeb (například oficiální validátor na adrese http://validator.w3.org).

2.3. DTD

DTD (Dokument Type Definition) je soubor podmínek, který definuje pravidla pro tvorbu určitého typu XML dokumentu. Lze pomocí něj definovat sadu platných značek a atributů v XML dokumentu. XML dokument pak bude validní, bude-li odpovídat nejen pravidlům XML, ale také danému DTD. Samotný soubor DTD je vlastně také XML dokumentem s příponou (.dtd).

DTD PRO XHTML

DTD existuje ve třech verzích: Strict, Transitional a Frameset.

- XHTML 1.0 Strict se používá u strukturovaných dokumentů osvobozených od formátovacích značek souvisejících s rozvržením stránky. Předpokládá se jeho užití společně s CSS, které umožní dosáhnout potřebných grafických efektů. Nicméně i tato verze obsahuje formátovací elementy, například <b> nebo <i> a naopak zavrhuje některé sémantické elementy, například <menu> [2].
- XHTML 1.0 Transitional je přechodným DTD pro webové stránky, který umožní používat překonané značky. Je vhodný pro formátování stránek vytvářených pro staré prohlížeče, které nerozumí kaskádovým stylům CSS nebo chcete-li používat ve svých dokumentech některé zavržené, ale sémantické elementy, například <menu> [2].
- XHTML 1.0 Frameset vám umožňuje používat zastaralé značky jako XHTML 1.0 Transitional a přidává podporu pro rámce. V dnešní době je dobré se mu vyhýbat použitím CSS nebo AJAXu [3].

2.4. CSS

Kaskádové styly (Cascading Style Sheets – CSS), v aktuální rozšířené verzi CSS3, byly vytvořeny W3C pro potřeby přesnějšího stanovení výsledného vzhledu internetových stránek. Díky stylům je možné dosáhnout výrazně lepších grafických efektů, které mnohonásobně překračující rámec možností jazyka HTML [2].

Používání kaskádových stylů ve srovnání se samotným HTML v praxi přináší tyto výhody:

- Rozsáhlejší možnosti formátování. Například pro formátování bloku textu – tj. určení vzdálenosti od jejich elementu či okraje stránky nenabízí HTML nic. CSS má vlastnosti padding a margin. V HTML by bylo potřeba vytvořit složitou konstrukci vnořených tabulek.
- Jednodušší údržba webové prezentace. Pokud chceme změnit nějaký detail, jako třeba barvu nadpisu, nemusíme složitě procházet HTML kód nebo různé HTML šablony, ale můžeme změnit pouze jednu vlastnost v CSS souboru, který je připojen ke všem stránkám [2].
- Oddělení struktury a stylu. V jednom (HTML) dokumentu budeme mít pouze sémanticky označen obsah a v druhém (CSS) dokumentu máme definice vzhledu stránek. Tím dosáhneme snadnějšího strojového zpracování samotného obsahu stránek, do kterého se nám nepletou prvky definující vzhled [2].
- Cachování stylů. Webový prohlížeč si může soubor se styly uložit do cache paměti, čímž může být dosaženo zrychlení načtení stránky. Na druhou stranu při použití externího CSS souboru dochází k dalšímu HTTP požadavku navíc oproti tomu, když bychom použili buď přímý zápis CSS nebo přímo formátování HTML.
- CSS vlastnosti jednotlivých elementů můžeme dynamicky měnit pomocí JavaScriptu.
- Z pomocí CSS můžeme jednoduše formátovat i jakýkoliv jazyk XML.
- Mohou také existovat různé styly pro různá výstupní zařízení. Jednoduše tak můžeme nadefinovat různý styl pro tisk, projekci, mobil, PDA, běžný prohlížeč či dokonce pro každý prohlížeč jiný styl. Specifikace CSS nezapomínají ani na zrakově postižené – je možno napsat styly pro hlasový syntetizátor nebo hmatovou čtečku Braillova písma [2].
- Koncový uživatel si může napsat svůj vlastní CSS styl pro libovolnou stránku. Většina prohlížečů nějakým způsobem podporuje uživatelské styly, takže uživatel si může například nastavit, aby měl všechny odkazy na všech webech vždy podtržené nebo aby na tomto konkrétním webu mělo písmo vždy černou barvu [2].

2.5. PHP

PHP (Hypertext Preprocesor : původně Personal Home Page tools) je skriptovací programovací jazyk vytvořený Rasmusem Lerdorfem v roce 1994, kdy vznikla binární část Common Gateway Interface (CGI). PHP se řadí mezi skriptovací jazyky interpretované na straně serveru, kdy je na straně klienta prezentována až výsledná akce. V roce 1997 s uvedením PHP 3 přišel významný zlom. Tato verze sice ještě nebyla plně použitelná v oblasti rozsáhlejších projektů, ale měla obrovské možnosti pro rozšíření a byla již vybavena velmi flexibilní API s mnoha databázovými protokoly. Koncem roku 1998, tedy nedlouho po uvolnění PHP3, začali Andi Gutmans a Zeev Suraski s přepracováním jádra PHP. Základním cílem bylo zvýšení výkonu u rozsáhlejších aplikací a zlepšení modularity jádra PHP. Nový engine splnil očekávání a po téměř dvou letech byl v roce 2000 uvolněn. Mezi nové klíčové funkce patřila široká podpora webových serverů, podpora HTTP sezení, výstupní buffery, bezpečnější

způsob zacházení s uživatelskými vstupy a několika novými jazykovými konstruktory. Po dlouhém vývoji pak v roce 2004 spatřilo světlo světa PHP 5. Hlavní změny spočívaly v následování světlových trendů a s tím spojené rozšíření objektově orientovaného programování v PHP. Dále pak PHP data objects extension pro optimalizaci práce s databázemi. V současné době se připravuje PHP 6, které bude definitivně implementovat plnou podporu Unicode [4], [5].

2.6. JAVA

Jedná se o objektově orientovaný programovací jazyk, který vyvinula společnost Sun Microsystems a uvedla v roce 1995 (po fúzi v roce 2010 je vývoj ve správě společnosti Oracle). Celosvětově dnes Java patří mezi nejrozšířenější programovací jazyky, a to i díky skutečnosti, že je od roku 2007 vyvíjena jako open source. Java je dnes součástí mnoha systémů a pracuje na širokém spektru platforem. Běžně se Javou setkáme v počítačích, mobilních telefonech, kreditních kartách, televizích, navigacích, atd. Dle informací, které uvádí společnost Oracle na stránkách java.com [6]:

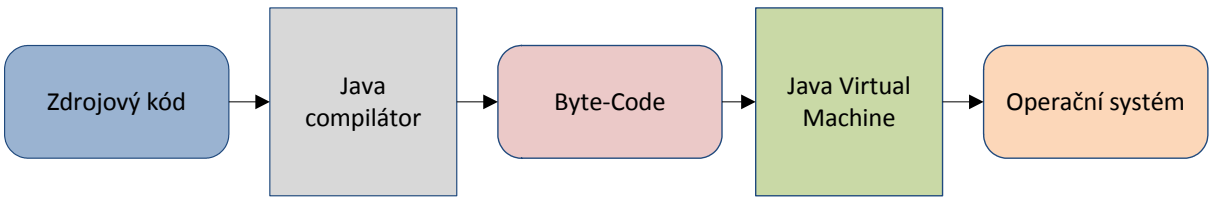
- Java běží na 1,1 miliardě stolních počítačů
- Každý rok je staženo 930 miliónů JRE
- Java běží na 3 miliardách mobilních telefonů
- 100% všech Blu-ray zařízení běží na Javě

Základní vlastnosti

- Syntax je odvozen od C/C++
- Objektově orientovaný jazyk
- Silně typovaný jazyky
- Nezávislý na architektuře
- Interpretovaný jazyk – na úrovni byte-code – Java Virtual Machine
- Multi-thread
- Dynamický

Java Virtual Machine

Java Virtual Machine, jak vyplývá z názvu, je virtuální prostředí, které běží jako proces v systému fyzického zařízení. Vstupním data pro JVM je „byte-code“ v podobě souboru (.class). Byte-code je přenositelný kód, který je přechodovou částí mezi zdrojovým kódem a strojovým kódem. Základní vlastností takového kódu je jeho přenositelnost mezi platformami. Teprve JVM provede překlad do strojového kódu dané platformy [7], [8].



OBRÁZEK 1 - ZPRACOVÁNÍ KÓDU V JAVĚ

Garbage collector

Jazyky jako C/C++ vyžadují, aby programátor prováděl ve svém vyvíjeném kódu ruční správu paměti. Každý objekt, který během běhu programu vzniká, musí být v případě, kdy již není potřebný, zrušen (zničen). Zrušením objektu dojde k uvolnění paměťových zdrojů, jež na sebe vázal. V případě, že by ke zrušení nedošlo, zůstala by paměť i nadále alokována a projevil by se „memory leak“, tedy situace, kdy program neúmyslně alokuje operační paměť, kterou již nedokáže uvolnit. Proto byl v Javě implementován „Garbage collector“. Jedná se proces správy paměti, který automaticky vyhledává v paměti objekty, aby vyhodnotil, zda jsou ještě zapotřebí. V pravidelných cyklech pak nepotřebné objekty ničí a uvolňuje tak paměť, kterou alokovaly. Vývojář se tedy nemusí o proces uvolňování paměti starat. Jistou nevýhodou je však skutečnost, že sám Garbage collector vyžaduje systémové prostředky, aby mohl provádět analýzu, zda je objekt ještě používán, nebo může být zrušen a jím alokované zdroje uvolněny [6], [8].

2.7. JAVASCRIPT

Hned v úvodu je třeba zdůraznit, že JavaScript nemá, kromě podobné syntaxe, nic společného s programovacím jazykem JAVA. Autorem JavaScriptu je Brendan Eich, který zahájil práci na interpretovaném objektově orientovaném skriptovacím jazyku, jehož by bylo možné použít na webu. V roce 1997 byl pak jazyk standardizován asociací ECMA, o rok později pak i ISO. Dnes je JavaScript běžně součástí zdrojového kódu stránek a stará se hlavně o interaktivnost obsahu a GUI. Sám skript je po stažení ze stránek interpretován až na klientském systému. S ohledem na tuto skutečnost však bylo nutné omezit operace, ke kterým má skript přístup. JavaScript například není schopen pracovat s lokálním souborovým systémem mimo operací s cookies a rovněž nemůže přistupovat k systémovým objektům. Všechny tyto omezení si kladou za cíl neohrozit soukromí uživatele, který má rovněž možnost zpracovávání JavaScript úplně zakázat [3].

3. OVĚŘENÍ IDENTITY UŽIVATELE - LDAP, RADIUS, SHIBBOLETH

V úvodu této kapitoly je třeba uvést termín Identity Management. Jedna z definic zní takto: *„Identity Management je sadou procesů, které umožňují autentizaci procesů náležících k identitě.“* [9]

Uvedený termín se tedy zabývá problematikou, jakým způsobem korektně propojit reálnou identitu s identitou virtuální (elektronickou). Do Identity Managementu patří tyto základní pojmy:

- **Identita** je soubor unikátních vlastností, které identitu jednoznačně definují
- **Identifikace** je proces, který ověří identitu vůči autoritě
- **Autorita** je nadřazený prvek v systému, který může rozhodnout o pravosti identity

V následující části budou dále upřesněný tři základní oblasti Identity Managementu.

AUTENTIZACE - OVĚŘENÍ IDENTITY

Co je to vlastně ověření identity? V oblasti výpočetní techniky se musíme bavit o procesu autentizace. Tento proces je určen k ověření deklarované identity vzhledem k určitému systému. V běžné praxi se nemusí jednat jen o ověření identity uživatele, ale i o ověření identity systému, procesu (služby), tedy obecně řečeno entity. Dnes je k dispozici cela řada metod ověřování identity.

- **Znalost** (typicky informace jako je jméno a heslo/pin)
- **Schopnost** (většinou se jedná o náhodnou kontrolní otázku např. 1+2=?, CAPTCHA, apod.)
- **Vlastnictví** (technický zabezpečovací prostředek token/dongle, SmartCard, certifikát, autentikátory)
- **Existence** (typicky biometrie tedy otisky prstů, hlas, oční duhovka)

Všechny výše popsané metody, snad vyjma poslední, jsou spolehlivé jen do chvíle, kdy je informace nutná k autentizaci kompromitována (prozrazena/ukradena). Zacházení s takovými údaji se proto musí řídit přísnými postupy. Je nejen nutné zabezpečit bezpečné předání informace vlastníkovi, ale i předání informace při samotné aplikaci, tedy autentizaci. Například pokud je autentizace založena na uživatelském jménu a heslu, je nutné zabezpečit, že se informace nebude mezi systémy pohybovat ve formě plain textu (prostý text), kterou je možno jednoduše odchytit.

V systémech, kde je nutná vyšší úroveň zabezpečení, se využívá multifaktorová autentizace, tedy kombinace několika výše popsaných metod. Dnes je multifaktorová autentizace běžná například v elektronickém bankovnictví, kdy je nutné použít například jméno a heslo ve spojení se zprávou SMS, nebo certifikát ve spojení s uživatelským jménem a heslem.

AUTORIZACE - NASTAVENÍ PRÁV

Jakmile dojde k úspěšné autentizaci, je nutné provést přiřazení práv, tedy autorizaci. Autorizace je proces, který přidělí autentizované entitě přístupová práva, která spravuje systém řízení přístupu (ACL). , jako jsou:

- RBAC (role-based access kontrol)
- LBAC (lattice-based access control)
- ABAC (attribute-based access kontrol)
- MAC (mandatory access kontrol)
- DAC (discretionary access kontrol)

AUDIT – SHROMAŽĎOVÁNÍ ZÁZNAMŮ

Audit je proces, kterým se shromažďují (logují) informace spojené s procesy autentizace a autorizace. Tyto záznamy jsou následně zdrojem informací při řešení systémových chyb, procesních a hlavně bezpečnostních incidentů (krádež identity).

3.1. LDAP

LDAP je odlehčená verze aplikačního protokolu DAP (Directory Access Protocol) odvozeného od standardu X. 500 pro přístup a správu dat na adresářových serverech. Pro zabezpečení přenášených dat je použit Framework SASL nebo GSSAPI (protokol Kerberos). Vývoj začal v roce 1993 a následně byl rozšířen v roce 1996 do aktuální verze LDAPv3. Zásadní rozdíl mezi implementací DAP a LDAP je ve filozofii klientského přístupu. Zatím co klient protokolu DAP komunikuje přímo s adresářovým serverem, komunikuje klient LDAP s adresářovou službou a až tato služba komunikuje s adresářovým serverem. Takto vzniklý interface umožňuje vývojářům vlastní implementaci adresářového serveru, který je skrze adresářovou službu přístupný standardními LDAP požadavky [10], [11].

OBRÁZEK 2 - KOMUNIKAČNÍ PROCES PROTOKOLU LDAP

LDAP se stal díky nízkým nárokům a jednoduché implementaci rychle populárním a položil základy následujícím internetovým protokolům jako XED, DSML, SPML a SLP [11].

STRUKTURA ADRESÁŘE

Data jsou uchovávána v hierarchické struktuře, které se říká adresář. Adresář obsahuje celkem čtyři datové modely.

- Informační model – definuje datové typy a základní informace (atributy) uložené v adresáři
- Jmenný model – definuje jak organizovat a odkazovat na data
- Funkční model – definuje operace v protokolu LDAP
- Bezpečnostní model – autentizuje a autorizuje přístup a manipulaci s adresářem

Každý model pak obsahuje jednotlivé specifické položky. Jednotlivé položky pak mají přiřazeny různé atributy. Atribut je pak vlastnost, kterou je možné naplnit daty.

ADRESÁŘOVÉ SCHÉMA

Celá hierarchie je uložena ve stromové struktuře s adresářovým schématem, které se říká Directory Information Tree (DIT). Na základě implementace informačního modelu vzniká schéma. Schéma v adresáři následně definuje všechny typy objektů a jejich atributy. Dále schéma umožňuje:

- Udržovat konzistenci dat
- Omezení duplicit
- Definovat pravidla – jaká data jsou v adresáři skladována
- Definovat interface pro přistup k datům
- Definovat podmínky při vkládání záznamů

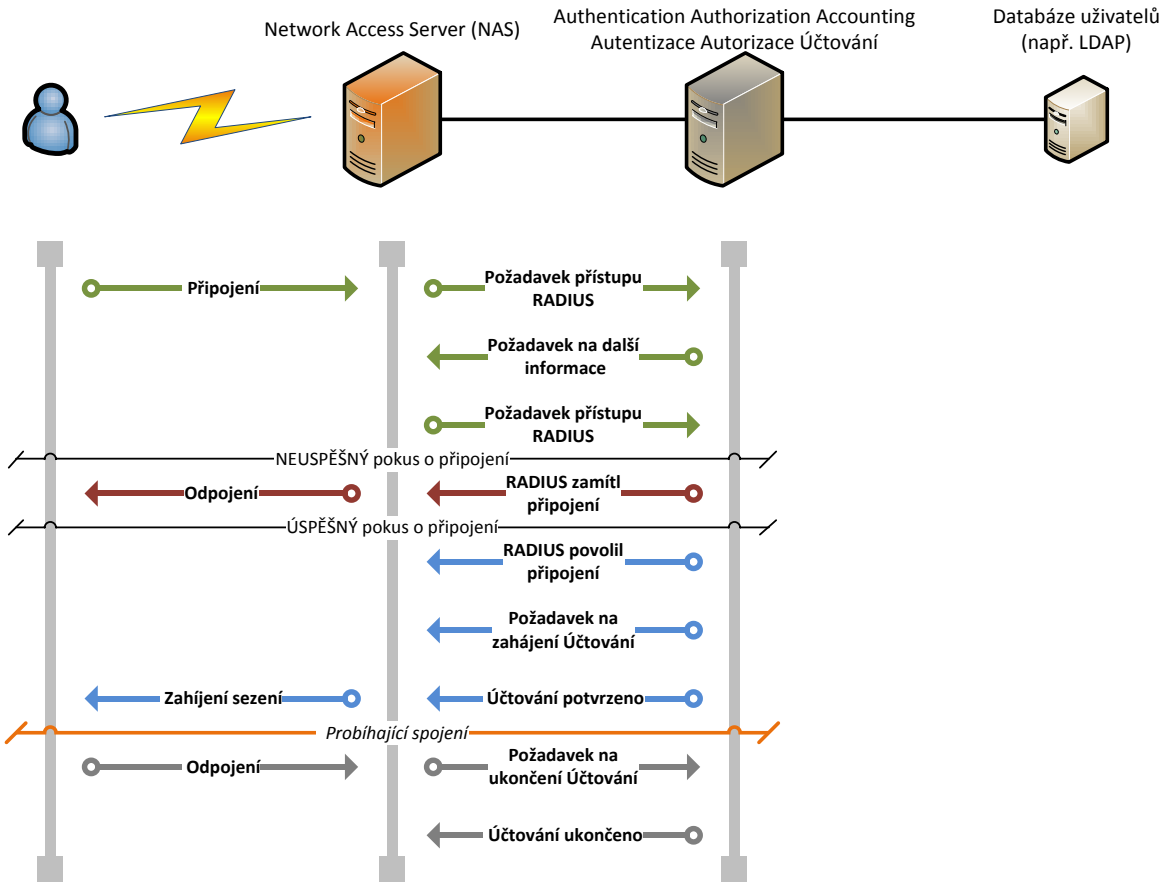
Každé schéma je možné při dodržení pravidel rozšířit tak, aby obsahovalo například další typy objektů. Třeba je možné definovat nové atributy pro potřeby vedení specifických záznamů o uživateli (atribut *gender* s možnými hodnotami *male*/*female*).

Dnes existuje celá řada implementaci adresářů využívajících aplikačního protokolu LDAP, ať komerční nebo nekomerční. Mezi nejznámější patří tyto:

- OpenLDAP
- Microsoft Active Directory
- Novell eDirectory
- IBM Lotus Domino

3.2. RADIUS

Je síťový protokol (IEEE 802.1X) na bázi komunikace klient/server pro autentizaci přístupu k síti vytvořený Lucent Remote Access, založený na protokolu AAA. Systém se běžně používá například při autentizaci připojení v rozsáhlých firemních nebo veřejných sítích (WiFi, ISP, apod.) [12].



OBRÁZEK 3 - SCHÉMA AAA

AUTENTIZACE

„Kdo jste?“

Server na základě klientem zaslaných informaci rozhodne, zda má klient přístup k požadovaným informacím (zda může být připojen). Informace nutné pro autentizaci server vyhodnocuje na základě pověření. Zdroj pověření je v dnešní době na RADIUS serveru zcela nezávislý. Většinou se jedná o databáze nebo adresářové služby jako je OpenID, SQL, Kerberos, LDAP, nebo AD [12].

AUTORIZACE

„Jaké služby vám mohu dát?“

Po úspěšné autentizaci RADIUS server přidělí navázanému sezení příslušná práva, která jsou podmínkou spojení s přístupem k sítovým zdrojům. Může se například jednat o nastavení šířky

přenosového pásma pří připojení k poskytovateli internetu (xDSL, WiFi, atd.) nebo začlenění do určitého subnetu [12].

ÚČTOVÁNÍ

„Co jste dělali se službami, když jste je používali?“

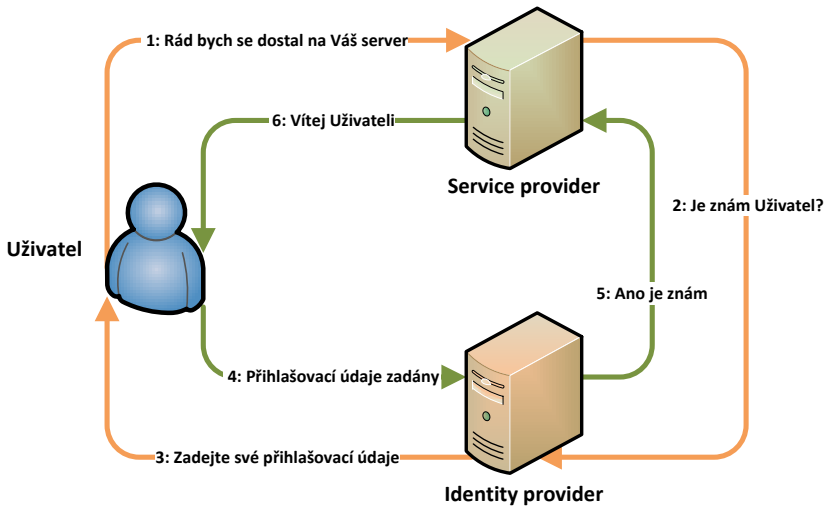
RADIUS server udržuje všechny záznamy o úspěšně zahájených sezeních. Následně ověřuje stav během probíhajícího sezení a zaznamenává ukončení sezení. Primárně jsou takto získaná data zdrojem podkladů pro potřeby vyúčtování poskytnutých služeb a statistiku [12].

PROTOKOL DIAMETR

Diametr (Diameter Base Protocol) je novou generací AAA protokolu vycházející z protokolu RADIUS. Jako nástupce protokolu RADIUS se snaží odstranit jeho hlavní deficity. Požadavky jsou kladeny především na zvýšení spolehlivosti (místo transportního protokolu UDP používá TCP/SCTP) a bezpečnosti ve spojení (zabezpečení na úrovni transportní vrstvy pomocí IPSEC nebo TLS). Dále má lepší zpětnou vazbu v případě chyby na straně serveru a je snadněji rozšiřitelný.

3.3. SHIBBOLETH

Je SSO (Single Sign-On) systém, který implementuje rozhraní pro komunikaci mezi poskytovatelem identity (identity provider) a poskytovatelem služby (service provider). SSO eliminují nutnost provádět opakovaně autentizaci do systémů, které mají společného poskytovatele identity jako je např. firemní/školní Adresářová služba. Shibboleth tedy poskytuje metody ověření pro přístup k poskytovateli služeb. V případě Shibbolethu se jedná o zabezpečené WWW stránky. Další aspekt, který Shibboleth řeší, je soukromí. Poskytovatel služby dostane skrze rozhraní jen informaci, zda se uživatel autentizoval proti poskytovateli identity a jaká obdržel práva (o právech může případně rozhodnou už poskytovatel služeb) [13].



OBRÁZEK 4 - SCHÉMA FUNKCE SHIBBOLETHU

Při prvním pokusu o přihlášení tedy dojde k přesměrování požadavku na poskytovatele identity. V případě úspěšné autorizace je vygenerován bezpečnostní token (ticket), který je poskytovatelem identity vrácen zpět na server, kam uživatel požaduje přístup. Pokud se uživatel následně pokusí přistoupit k dalšímu systému, který využívá stejného poskytovatele identity a je-li bezpečnosti kontext stále platný, bude uživateli opět zpřístupněn obsah.

Systémy využívající SSO však vyžadují důsledné dodržování základního bezpečnostního principu, tedy odhlašování (korektní ukončení sezení). V případě, že uživatel nechá jeden systém přihlášený, má případný útočník automaticky přístup i k systémům ostatním.

4. NÁVRH DOKUWIKI PRO LABORATOŘ IP TELEFONIE

V první části této kapitoly je stručně popsána historie Wiki a uvedeno několik nejrozšířenějších Wiki systémů. V druhé části je pak věnována pozornost samotnému návrhu implementovaného Wiki systém – DokuWiki, pro Laboratoř IP Telefonie na Katedru telekomunikační techniky (dále jen kat440).

4.1. CO JE WIKI

Systém Wiki původně vznikl pro podporu komunity programátorů zabývajících se problematikou návrhových vzorů. Základní požadavek byl, aby systém fungoval jako diskuze, tedy připívat může kdokoliv, ale navíc s možností následné editace a historie změn. A tak se zrodil vlastní jednoduchý značkovací jazyk „wikitext“ pro rychlou a přehlednou tvorbu obsahu a systém pro správu takto vytvářeného obsahu „wiki“. Na počátku stál projekt Portland Pattern Repository implementovaný Wardenem Cunninghamem roku 1995. Tento člověk je zároveň odpovědný za společný název „wiki“, „wiki-systémy“ nebo dnes také „wiki engine“. Samo označení wiki plně vystihuje klíčovou vlastnost a tou je rychlost, ve smyslu rychlé výměny informací. Cunningham se pro název nechal inspirovat u hromadné autobusové dopravy na letišti v Honolulu. V havajštině totiž výraz „Wiki Wiki“ znamená „rychle“ – „velmi rychle“.

V dnešní době existuje celá řada wiki systémů tzv. „wiki engienů“ od jednoduchých, až po velmi sofistikované, kdy některé projekty dosáhly velké modularity a disponují vlastním API pro podporu komunity uživatelů a vývojářů. Komunity dnes také stojí za rozsáhlými bázemi modulů a pluginů, které mnohdy výrazně rozšiřují funkce wiki systému. Tento vývoj, který se převážně děje v oblasti „open source“, ale zapříčinil rozkol v implementaci „wikitextu“ a jednotlivé vývojové týmy začaly definovat vlastní značky, nebo původním značky významově pozměnili či vůbec nepoužili.

NEJZNÁMĚJŠÍ WIKI SYSTÉMY

Existuje kolem 100 různých wiki systémů. V této skupině dnes existuje asi 10 nejrozšířenějších, z nichž jsou nejzajímavější tyto:

MediaWiki

Mezi neznámějšími wiki enginy se dnes suverénně na prvním místě drží, a to dle uveřejněných wiki stránek, systém MediaWiki. Hlavním důvodem je zde všem obecně známá otevřená encyklopedie Wikipedia, díky, které MediaWiki vznikla. Historie MediaWiki je dnes vlastně historií Wikipedie, která se začala psát v roce 2001. Projekt tehdy vznikl na podporu již existujícího projektu Nupedie a byl postaven na systému UseModWiki. Vzhledem k nepopularitě Nupedie, kterou zapříčinila nutnost schvalování publikovaných článků, jenž odradila velkou část potencionálních přispěvatelů, došlo ke zrušení Nupedie a převedení stávajícího obsahu do otevřené Wikipedie.

Již v roce 2001 zahájil student Magnus Manske práci na novém enginu postaveném na PHP skriptu s využitím MySQL databáze. Zároveň byly definovány mechanizmy, jako jmenný prostor pro organizaci stránek, a speciální stránky jako seznam přispěvatelů nebo uživatelské sledování změn na stránkách. Následně pak v roce 2003 založil Jimmy Wales nadaci Wikimedia pro podporu vývoje a produkt byl pojmenován MediaWiki. V současné době již usazený projekt pokračuje ve vývoji, ovšem v relativně malých krocích.

UseModWiki

Je jeden z nejstarších wiki systému a je psán v jazyku Perl. Jako většina wiki systémů ukládá wiki stránky pouze v souborové databázi bez podpory databáze relační. Od 15. ledna 2001 do začátku roku 2002 byl systém používán i pro chod Wikipedie, kde byl následně nahrazen MediaWiki. Protože tento systém stál u zrodu MediaWiki, mají oba systémy mnoho společných rysů. V současné době vývoj stagnuje, ale projekt je pořád živý.

MoinMoin

Je engine psaný v jazyku Pythonu a může být provozován jak na souborové, tak relační databázi. Projekt sám staví na enginu PikiPiki

PmWiki

Tento systém psaný v PHP má možnost využít jak ukládání do souborů, tak do MySQL. Zajímavostí je ochrana jednotlivých wiki stránek heslem. Tato skutečnost umožňuje vysokou míru editovatelnosti při zachování dobré ochrany proti vandalismu a spamu, který jinak většinu otevřených wiki systémů postihuje.

DokuWiki

Autorem je Andreas Gohr, který zahájil vývoj v roce 2004. Celý systém je implementován v PHP s rozsáhlou API pro tvorbu pluginů. Vývoj systému je zacílen na privátní segment, malé a střední společnosti a týmy, a to za účelem vedení dokumentace a blogování. Staví čistě na ukládání stránek do adresářových struktur v podobě plain textový souborů. Sami vývojáři považují tento způsob ukládání za jednu z hlavních výhod wiki systému. Snadný přístup ke stránkám na úrovni hostujícího operačního systému ve spojení se snadnou úpravou, zálohováním a případnou fyzickou migrací. DokuWiki používá podobný wikipage syntax jako MediaWiki, ale oproti MediaWiki má celý systém podstatně menší systémové nároky vzhledem k segmentu, na který cílí.

4.2. NÁVRH ŘEŠENÍ

Koncept nasazení DokuWiki pro Katedru telekomunikací (kat440) jsem rozdělil do následujících bodů ve formě otázky.

- Jakou zvolit systémovou platformu a webový server?
- Jakou verzi DokuWiki implementovat a proč?
- Jaký způsob autentizace DokuWiki podporuje a jaký zvolit?
- Jak vhodně navrhnout hierarchie jmenných prostorů?
- Jaké použít v DokuWiki uživatelské pluginy?
- Jak a kdy zálohovat data uložená v DokuWiki?

4.3. SYSTÉMOVÁ PLATFORMA

Nutným požadavkem pro běh DokuWiki je webový server, který podporuje PHP, a to ve verzi minimálně 5.1.2. Mezi tyto web servery patří Apache, IIS, lighttpd, Nginx a další. Protože je celá řada web serveru multiplatformních, záležela volba operačního systému na zvoleném webovém serveru. Vzhledem k optimalizaci zdrojů, obecnému rozšíření a dobrou podporu byl zvolen web server Apache a operační systém Linux, distribuce Debian. Aby byl systém adekvátně zabezpečen, hlavně při autentizaci uživatele, je celý výsledný web přístupný pouze přes protokol HTTPS. Web server dále využívá všech možností zabezpečení pomocí (.htaccess) společné s aplikací restrikcí pro chod a intepretaci PHP.

4.4. VERZE DOKUWIKI

Při volbě verze DokuWiki bylo třeba brát v potaz následnou kompatibilitu s pluginy, které využívají API DokuWiki. Některé pluginy již totiž nejsou dlouhodobě udržovány a testovány. Na novějších verzích DokuWiki se tedy může projevit nekompatibilita, a proto nemusejí korektně fungovat, nebo mohou narušovat chod jiných pluginů a dokonce cele DokuWiki. Byla tedy provedena celá řada testovacích instalací, které si kladly za cíl vybrat nejvhodnější a přitom nejaktuálnější verzi DokuWiki. V repositáři distribuce Debian se nachází schválené verze 2009-12-25c „Lemming“. V současné době však existují další tři aktuálnější stabilní verze.

- 2012-01-25 „Angua“
- 2011-05-25 „Rincewind“
- 2010-11-07 „Anteater“

Důvod, proč repositář obsahuje jen výše uvedou verzi, je dán filozofií distribuce Debian. Debian sice patří mezi nejrozšířenější serverové distribuce, ale zároveň mezi distribuce nejkonzervativnější. Publikování aktuálnějších verzí prochází složitým a časově náročným testováním, které v mnoha případech také odstraní z aplikací nebezpečné materiály, na základě politiky Debianu (v případě DokuWiki se jedná třeba o Adobe Flash aplikaci pro nahrávání medií). Debian verze, také obsahuje změnu v hierarchii dat instalované DokuWiki. Jedná se o jednu ze skutečností dle politiky Debianu.

DokuWiki	Výchozí cesta
Kořen	/usr/share/dokuwiki/
Data	/var/lib/dokuwiki/data/
Konfigurace	/etc/dokuwiki/

TABULKA 1 - VÝCHOZÍ CESTY PRO DOKUWIKI Z REPOSITÁŘE DEBIANU

Během testování a na základě zkušeností uživatelů uveřejněných na stránkách ***dokuwiki.org***, se jako nevhodnější verze DokuWiki ukázala 2010-11-07a „Anteater“ instalovaná se zdrojového balíčku přímo ze stránek DokuWiki. Protože se jedná o verzi přímo od vývojářů, probíhá její instalace (bez modifikace) do výchozí cesty /var/www/.

4.5. AUTENTIZACE

DokuWiki umožňuje použití celé řady autentizačních metod tzv. “backend” [14]. Jejich výběr závisí na účelu použití a prostředí, v němž je DokuWiki nasazen. Přímo v DokuWiki je implementováno pět metod, samozřejmě existuje celá řada rozšiřujících mechanizmů, dodatečně vyvinutých komunitou.

Metoda	Popis
AD	Autentizace oproti Active Directory
LDAP	Autentizace oproti LDAP adresáři
MySQL	Autentizace oproti MySQL databázi
PGSQL	Autentizace oproti Postgres databázi
PLAIN	Výchozí mechanizmus, který používá plaintextové soubory

TABULKA 2 - ZÁKLADNÍ METODY AUTENTIZACE V DOKUWIKI

V prostředí sítě VŠB-TU se pro všechny uživatelské účty používá jednotný LDAP adresář. Pro autentizaci byl proto zvolen „backend“ LDAP.

Dále se dané problematice věnuje kapitola 5.

4.6. PLUGINY

Základní editační funkce DokuWiki plně dostačují pro tvorbu běžného obsahu. Protože však má implementace sloužit pro vedení technické dokumentace, bylo třeba rozšířit editační a výstupní možnosti o nové funkce. Bylo tedy vybráno a otestováno několik pluginů, které umožňují zlepšit přehlednost uložených dokumentů, umožnit přesný typografický výstup a umožnit zvýraznění syntaxe zdrojového kódu vkládaného do stránek. Příklady použití jednotlivých pluginu i s příklady jsem umístil přímo v DokuWiki, a to ve zvláštní sekci „How-to“. Pro přehlednost zde jen uvedu seznam a význam použitých pluginů.

CSV Plugin

Slouží pro generování tabulek ze souboru (.csv) uloženého ve jmenném prostoru. Plugin je vhodný hlavně v případě, kdy je nutné uveřejnit vetší množství opakovaně generovaných dat (například záznam měření).

Folded plugin

Plugin umožňuje složit (svinout) části obsahu na stránce. Při vhodném použití pluginu se výrazně zvyšuje přehlednost u rozsáhlejších stránek. Vhodné využití nalezne například u vysvětlivek nebo ukázek zdrojového kódu.

LaTeX plugin

Umožňuje generovat výstup ve vysoké typografické kvalitě a zachovat tak přesnost podání. Vhodný je především pro přesné citace.

mathpublish Plugin

Funguje obdobně jako LaTeX, ale je primárně zaměřen na generování matematických vzorců. Pro tyto účely má i přehlednější syntax.

NSpages plugin

Tento plugin je určen k tvorbě dynamického obsahu. Plugin umí z definovaného jmenného prostoru generovat přehlednou tabulku obsahu. Generovaný obsah lze definovat na základě parametrů.

SyntaxHighlighter3 Plugin

DokuWiki nativně podporuje zvýrazňování syntaxe zdrojového kódu díky projektu GeSHi (Generic Syntax Highlighter). Plugin SyntaxHighlighter3 je implementován jako alternativa.

Video Share Plugin

Plugin umožňuje integrovat videa z nejznámějších služeb jako je třeba YouTube. V How-to je rovněž uvedeno jak pracovat s videm ve formátu (.swf a .flv), které jsou uloženy přímo v DokuWiki.

Google Chart Plugin

Plugin využívá API „Google Char“ s jehož pomocí dokáže generovat různé grafy.

Barcode - QR Code

Dnes velmi rozšířená metoda pro rychlou výměnu informaci pomocí QR kódu. Plugin umí jednoduše generovat kód na základě zadaných informací.

Note

Jednoduchý plugin, který zvýrazní danou část textu jako noticku.

4.7. HIERARCHIE JMENNÝCH PROSTORŮ

Namespace, neboli jmenný prostor, je označení adresáře(ů), který slouží pro třídění obsahu Wiki stránek. DokuWiki nepoužívá pro uchovávání obsahu relační databázi, ale adresáře a textové soubory na úrovni systému. Jmenné prostory i dokumenty se pojmenovávají bez použití diakritiky, zvláštních

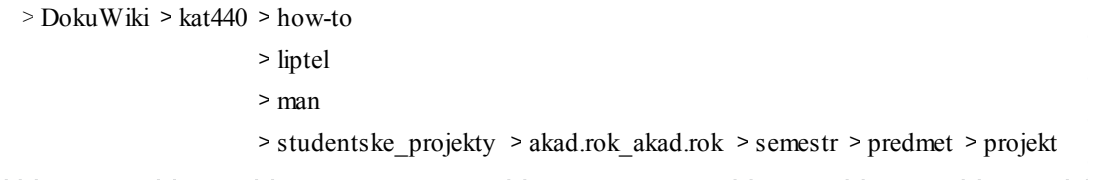
znaků a mezer (ta se nahrazuje podtržítkem). Aby nedocházelo k vytváření prázdných struktur, nelze v DokuWiki vytvořit prázdný jmenný prostor. Každý jmenný prostor musí obsahovat buď jiný jmenný prostor, nebo dokument. Následné rušení jmenných prostorů probíhá v opačném gardu. Pokud jmenný prostor neobsahuje další jmenný prostor nebo stránku, tak jej systém odstraní a tím jmenný prostor zaniká. Stejné podmínky platí i pro stránky, pokud stránka neobsahuje žádná data, automaticky zaniká. Hierarchie stránek se uchovává v datovém adresáři DokuWiki v podadresáři „pages“, paralelně však existuje na úrovni adresáře „pages“ adresář „media“. V případě, že je do jmenného prostoru nahrán soubor (obrázek, dokument, atd.), není uložen ve struktuře adresářů „pages“, ale ve struktuře adresářů „media“. Struktura vedoucí k jmennému prorostou, kam má nahraný soubor náležet, je tak zrcadlena do struktury „média“. Obě výše popsané hierarchie však nejsou vzájemně synchronizovány. V případě, že zaniká jmenný prostor v hierarchii „pages“, nezanikne jmenný prostor obsahující dokument v hierarchii „media“ a opačně, zanikne-li dokument v hierarchii „media“, nezanikne příslušný jmenný prosto v „pages“. V případě adresáře „media“ je tedy nutné dbát na odstraňování již nepotřebných dat.

V následující tabulce je přehledně uveden celý obsah datového adresáře DokuWiki. Vzhledem k faktu, že při vzniku dokumentu dochází k inicializaci vnitřních systémových akcí jako generování cache, indexu, metadat a historie, je velmi náročné a nebezpečné manipulovat z již existujícími stránkami a jmennými prostory v hierarchii „pages“. V případě, že k takové manipulaci dojde (na úrovni souborového systému), dochází k nekonzistenci a narušení funkcí jako je vyhledávání, historie stránek a atd.

Adresář	**Obsah**
data/attic	Obsahuje staré revize stránek (historii změn)
data/cache	Obsahuje cache parsovaných stránek
data/index	Obsahuje indexy pro rychlé vyhledávání
data/locks	Obsahuje zámky právě editovaných stránek
data/media	Obsahuje hierarchii nahraných medií (obrázky, dokumenty, atd.)
data/meta	Obsahuje metadata stránek
data/pages	Obsahuje hierarchii stránek
data/tmp	Obsahuje dočasné soubory

TABULKA 3 - OBSAHY ADRESÁŘŮ V DATOVÉM ADRESÁŘI DOKUWIKI

Vzhledem k omezení bylo nutné navrhnout strukturu, která nebude vyžadovat následnou manipulaci se jmennými prostory a dokumenty. Na základě podkladů a požadavků od vedoucího bakalářské práce byla struktura navržena takto:



OBRÁZEK 5 - ZÁKLADNÍ SCHÉMA DOKUWIKI PRO KAT440

Jmenný prostor kat440 tedy slouží jako kořen pro celou katedru. V případě, že by v DokuWiki měla vzniknou sekce, která přímo nesouvisí s katedrou, je zde tedy možnost umístit ji do úplného kořene celé DokuWiki.

Jmenný prostor	**Náplň**
how-to	návody a postupy pro DokuWiki a pluginy
liptel	laboratoř IP telefonie
man	technická dokumentace pro správce DokuWiki serveru
studentske_projekty	kořen pro vedení studentských projektů

TABULKA 4 - NÁPLŇ JEDNOTLIVÝCH JENNÝCH PROSTORŮ

Nyní se blíže podíváme na strukturu jmenného prostoru „*studentske_projekty*“. Hierarchie se zde snaží maximálně přiblížit struktuře akademického roku. Každý nový podřízený jmenný prostor s označením aktuálního akademického roku má v sobě dva jmenné prostory s označením semestru. Každý semestr pak obsahuje jmenné prostory jednotlivých předmětů a následně každý předmět obsahuje jmenné prostory pro jednotlivé projekty. Počet předmětů je v letním semestru pět a v zimní dva. Každý předmět pak počítá s cca pěti projekty. Výsledná hierarchie plně pokrývá všechny předpoklady na funkci, ale její tvorba by byla za použití standardních postupů (ručně) velmi zdlouhavá.

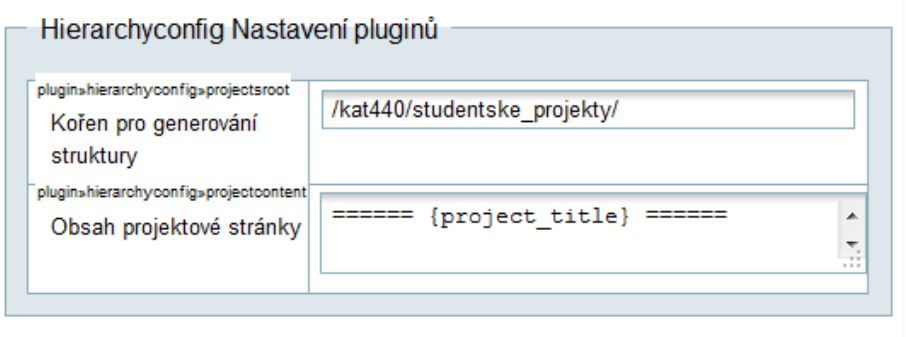
4.8. NÁVRH PLUGINU KONFIGURACE HIERARCHIE

Původní koncepce pluginu, který by umožnoval automatický generovat jmenné prostory, počítal z využitím pluginu Virtual group (viz. kapitola Autentizace). Propojení obou funkčností mělo umožnit nejen snadné generování projektových struktur, ale i zajistit přiřazování přístupových práv jednotlivých bezpečnostních skupin, dle ACL, určeným projektům. Během vývoje jsem však narazil na problémy s mechanizmem funkčnosti celé DokuWiki. V dokumentaci popsané akce (triggery) se mi nepodařily propojit, aby bylo možné dynamicky načítat seznamy uživatelských LDAP loginů z dokumentů umístěných ve jmenných prostorech jednotlivých projektů. Proto se vývoj pluginu zaměřil jen na generování struktury studentských projektů [Příloha CD/DVD].

POPIS PRÁCE S PLUGINEM

Konfigurace pluginu a jeho ovládání je umístěno v administrátorské sekci DokuWiki „Správa“. Hlavní nastavení pluginu je v menu „Správa nastavení“ sekce „Hierarchyconfig Nastavení pluginů“.

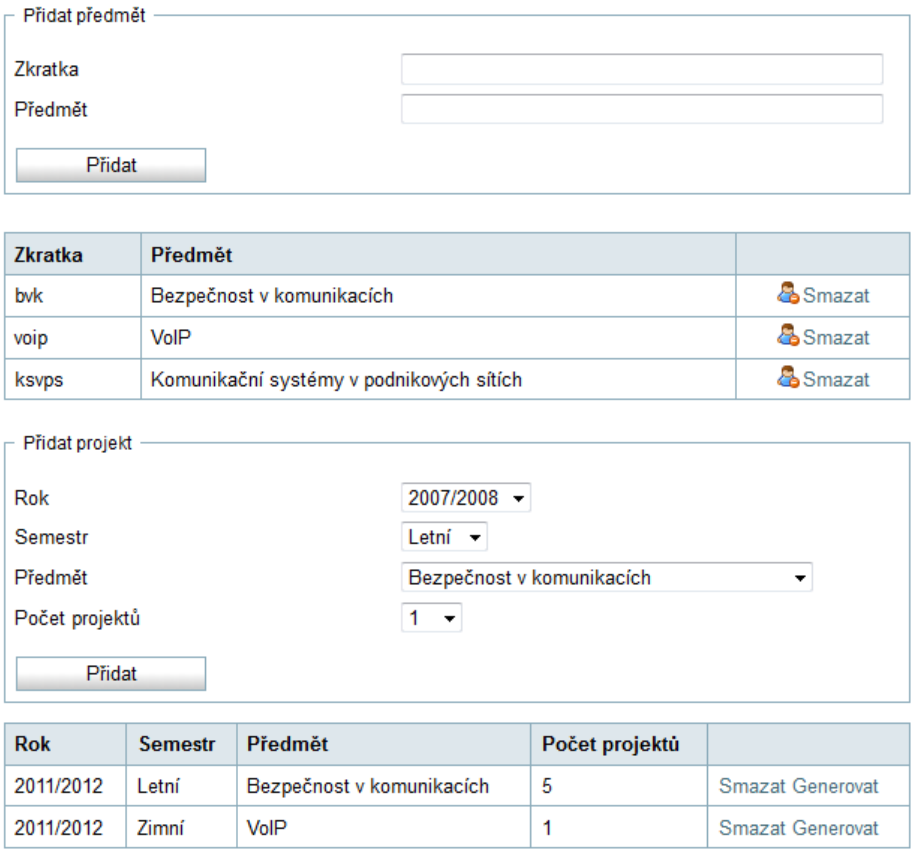
Zde je možné nastavit výchozí kořen, do kterého plugin generuje hierarchii a tělo dokumentu společného pro všechny projekty.



OBRÁZEK 6 - SPRÁVA NASTAVENÍ HIERARCHY PLUGINU

Obsah projektové stránky a její výsledný vzhled je v [Příloha A]

Nastavení pro přípravu a generování hierarchie je v sekci „Další pluginy“ a plugin „Konfigurace hierarchie“.



OBRÁZEK 7 - ROZHRANÍ PRO PŘÍPRAVU A GENEROVÁNÍ

V sekci „Přidat předmět“ se určí jaká je zkratka předmětu a přesný název předmětu. Po přidání se objeví v tabulce. V následující sekci „Přidat projekt“ se vybere příslušný akademický rok, semestr, název předmětu a počet projektů. Po přidání se objeví v následující tabulce. Výsledné vygenerování struktury provede na každém řádku funkce „Generovat“. Funkce „Smazat“, v obou případech

odstraňuje pouze záznamy uložené v pluginu, nedojde tedy ke smazání žádných jmenných prostorů ani dokumentu v DokuWiki.

Jednotlivá nastavení předmětů jsou v podobě serializovaného pole ukládána v datovém kořenu DokuWiki ve formě dvou souborů predmety.php a projekty.php.

4.9. ZÁLOHOVÁNÍ

Vzhledem ke skutečnosti, že jsou data v DokuWiki ukládána v podobě textových souborů, je zálohování možné přímo na úrovni souborového systému. Pro systematické zálohování tedy stačí skript, který v pravidelných cyklech provádí zálohování datové struktury na externí úložiště. Pro archivování je použit formát (.tar), který je schopen uchovat nejen archivovaná data, ale zachovává i informace souborového systému (vlastníka, skupiny, atd.). Aby byl výsledný archiv menší, bude ještě celý komprimován metodou bzip2. Pro implementaci takového řešení byl použit skriptovací jazyk BASH. Jako výchozí byl použit mechanizmus popsaný na stránkách projektu DokuWiki [15]. Zálohy se budou provádět na externí úložiště NAS, které je dostupné na Katedře telekomunikační techniky.

Skript provádí zálohování do třech samostatných archivů dle obsahu.

Zálohované adresáře			
Cesta	Detailní definice obsahu		
	wiki	*data*	*media*
/var/www/wiki/	bin conf inc lib	data/attic data/locks data/meta data/pages data/tmp	data/media

TABULKA 5 - TABULKA ZÁLOH A JEJICH OBSAHU

Název	**Popis**
wiki	konfigurace DokuWiki bez dat
data	data mimo adresář media
media	adresář media

TABULKA 6 - STRUČNÁ DEFINICE ZÁLOH

Po archivaci zašle skript email s informací o provedených zálohách, jejich velikostech a celkové době trvání zálohy. Informace o plánování a přesném umístění záloh jsou popsány v kapitole 6. Všechny parametry jsou ve skriptu vysvětleny formou komentářů [Příloha B], [Příloha CD/DVD].

5. AUTENTIZACE PŘÍSTUPU

Primární autentizace je na celé VŠB-TU postavena na Adresářové službě Novell eDirectory v8.8. Tato verze prostřednictvím agentů implementuje i podporu pro protokoly SASL (LDAPv2 a LDAPv3) a GSSAPI(Kerberos). V současnosti je s ohledem na bezpečnost nutné pro využití služeb adresáře používat LDAPS, tedy spojení s využitím SSL přes TCP port 636. Jazyk PHP díky rozšíření „php_ldap“ nemá s tímto požadavkem problémy. Výchozí adresa serveru je „ldaps://ldap.vsb.cz:636“. Přístup k adresáři je možný jak anonymně, tak s přihlášením. Rozdíl je jen v úrovni informací a funkcí, které jsou následně zpřístupněny. Dle informací od CIT-IS je možná autentizace i v anonymním režimu, kdy jsou ale následně dostupné pouze základní informace o uživatelích, skupinách a jednotlivých členstvích. Pro správnou funkci DokuWiki je klíčová dostupnost informace o loginu, celém jménu a emailu. Abych si ověřil dostupnost těchto informací i v anonymním režimu, byl použit příkaz „ldapsearch“. Na serveru s DokuWiki však bylo ještě nutné modifikovat „ldap.conf“.

```
URI ldaps://ldap.vsb.cz:636
TLS_REQCERT never
```

Druhý parametr je podmíněn vydáním certifikátu pro server. Následně byl proveden dotaz:

```
ldapsearch -H ldaps://ldap.vsb.cz -x cn=kri003 fullName uid mail
```

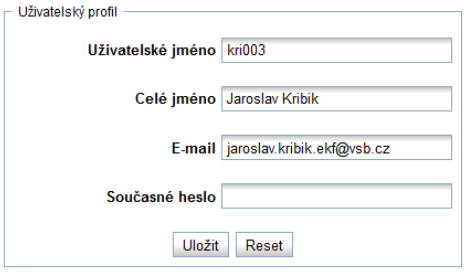
Dotaz vrátil:

```
# kri003, 3, USERS, VSB
dn: cn=kri003,ou=3,ou=USERS,o=VSB
mail: jaroslav.kribik.ekf@vsb.cz
uid: kri003
fullName: Jaroslav Kribik
```

Na základě tohoto úspěšného testu byla provedena konfigurace autentizace v DokuWiki. Všechny konfigurační soubory jsou uloženy ve „/var/www/wiki/conf“. V konfiguračním souboru local.php byl změněn parametr $conf['authtype'] „plain“ na „ldap“. Následně byl v adresáři „conf“ vytvořen nový soubor „local.protected.php“. Rozdíl mezi oběma soubory spočívá ve skutečnosti, že „local.php“ je možné editovat pomocí webového administrátorského rozhraní v DokuWiki a při změně je generován. Parametry, které jsou dále pro nastavení autentizace nutné, ale menu neobsahuje a při manipulaci administrátora s konfigurací přes webové rozhraní by byly automaticky smazány. Konfigurace musí obsahovat adresu serveru, kontext organizační jednotky s uživateli, filtr pro vyhledávání uživatelů dle typu objektu. Poslední parametr pouze mapuje atribut fullname do hodnoty name v DokuWiki, aby bylo korektně zobrazováno celé jméno.

```
/* Konfigurace pristupu k LDAP */
$conf['auth']['ldap']['version'] = '3';
$conf['auth']['ldap']['server'] = 'ldaps://ldap.vsb.cz:636';
$cons['auth']['ldap']['usertree'] = 'ou=USERS,o=VSB,dc=ldap,dc=vsb,dc=cz';
$conf['auth']['ldap']['userfilter'] = '(&(objectClass=person)(&(objectClass=ndsLoginProperties)(commonN
ame=%{user})))';
$conf['auth']['ldap']['mapping']['name']  = 'fullname';
```

Po této úpravě již jen byla restartovaná služba webového serveru a provedeno přihlášení k DokuWiki pomocí školního loginu a hesla. Po úspěšném přihlášení jsem provedl kontrolu načtení požadovaných dat přes „Upravit profil“.



OBRÁZEK 8 - INFROMACE O PŘILÁŠENÉM PROFILU

5.1. AUTORIZACE

Pro potřeby autorizace je však adresář nevhodný. DokuWiki sice dokáže pracovat se skupinami uživatelů v adresáři, ale stává se tak závislá na struktuře, kterou není možné z DokuWiki jakkoliv ovlivnit. V případě použití skupin přes adresář by manipulace z účty/skupinami v adresáři vždy vyžadovalo asistenci správce adresářového serveru. Sama DokuWiki však podporuje správu uživatelů pouze v případě, kdy je použita výchozí metoda autentizace, tedy PLAIN. Pokud je aktivována jiná metoda, v našem případě LDAP, stane se celá struktura správy uživatelů v DokuWiki nedostupnou. Bylo tedy třeba nalézt možnost, jak využívat adresář pro autentizaci, ale autorizaci provádět na úrovni DokuWiki přes interní ACL.

OBRÁZEK 9 - ROZDÍL VE SPRÁVĚ V REŽIMU PLAIN A LDAP

SPRÁVA PŘÍSTUPOVÝCH PRÁV UŽIVATELE

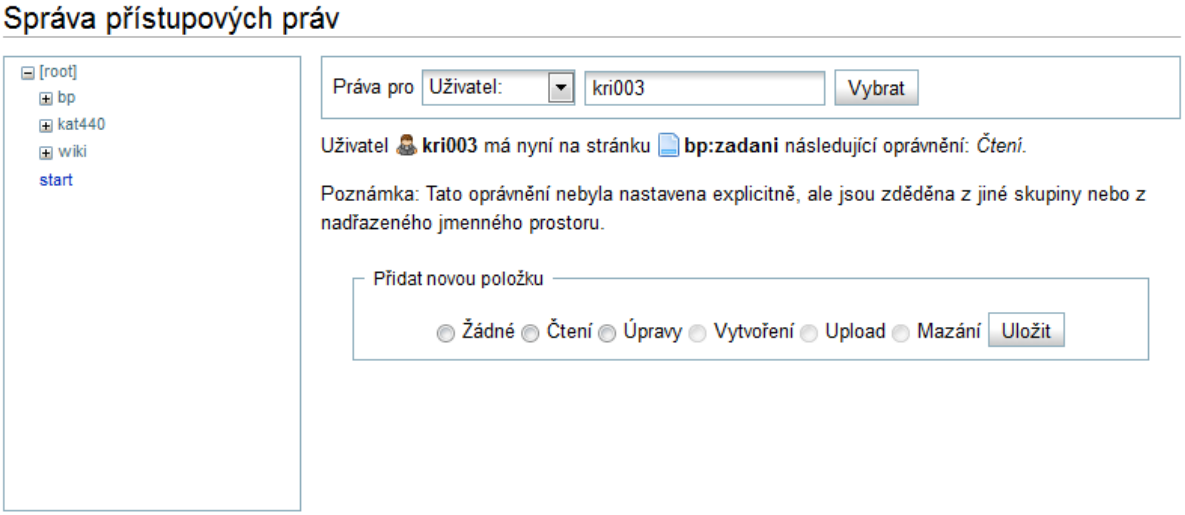
Tato sekce v administrátorském menu umožňuje definovat práva uživatelů a skupin k jednotlivým jmenným prostorům a dokumentům. V případě, že je použit k autentizaci LDAP, stačí pro přiřazení práv konkrétnímu uživateli znát jeho login.

První přiřazení práv – uživatel není uveden v seznamu ACL

V roletce se vybere položka „Uživatel“, v následujícím poli se vyplní login např. „kri003“. Po stisku tlačítka „Vybrat“ dojde k zavedení uživatele. Následně již stačí jen vybrat jmenný prostor a definovat příslušná práva.

Opakované přiřazení práv – uživatel je již uveden v seznamu ACL

V roletce se může vybrat přímo konkrétního uživatel nebo se může opakovat stejný postup jako při prvním přiřazení práv uživateli.



OBRÁZEK 10 - PŘIŘAZENÍ PRÁV UŽIVATELI NA ZÁKLADĚ LOGINU

SPRÁVA PŘÍSTUPOVÝCH PRÁV SKUPINY

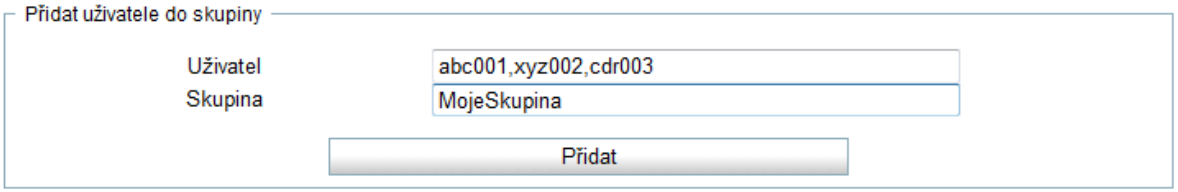
Problém však nastane v případě, kdy chceme pracovat se skupinami, tedy definovat v DokuWiki skupinu, které přiřadíme práva a následně již jen modifikovat její členy.

5.2. ŘEŠENÍ PROBLÉMU POUŽITÍ SKUPIN - PLUGIN VIRTUALGROUP

Pro vyřešení tohoto problému byl použit plugin Virtualgroup, který umožňuje definovat bezpečnostní skupiny a do těchto skupin začleňovat loginy z adresáře. Následně je možné skupinu použít ve „Správě přístupových práv“ a definovat jí práva.

PRÁCE S PLUGINEM VIRTUAL GROUP

Nastavení pluginu je v administrátorské sekci „Správa“ v položce „Další pluginy“ pod názvem „Virtuální skupiny“. Plugin byl upraven, aby akceptoval hromadné přidávání účtu, kdy jsou jednotlivé účty odděleny (čárkou nebo středníkem).



OBRÁZEK 11 - PŘIDÁNÍ UŽIVATELE(Ů) DO SKUPINY

Po provedení akce „Přidat“ jsou jednotlivá začlenění zobrazena v tabulce, kde je lze následně editovat a mazat.

Uživatelé	Skupiny	
abc001	MojeSkupina	Editovat • Smazat
xyz002	MojeSkupina	Editovat • Smazat
cdr003	MojeSkupina	Editovat • Smazat

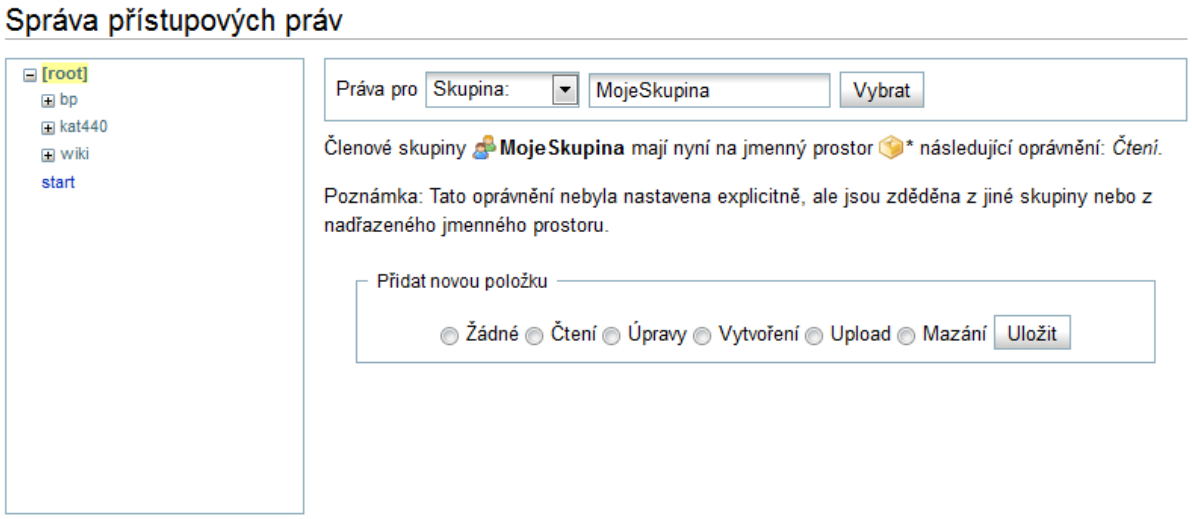
OBRÁZEK 12 - VÝPIS ČLENSTVÍ

První přiřazení práv – skupina není uveden v seznamu ACL

V roletce se vybere položka „Skupina“, v následujícím poli se vyplní název skupiny tak, jak byla definována přes Virtualgroup např. „MojeSkupina“. Po stisku tlačítka „Vybrat“ dojde k zavedení skupiny. Následně již stačí jen vybrat jmenný prostor a definovat příslušná práva.

Opakované přiřazení práv – skupina je již uveden v seznamu ACL

V roletce se může vybrat přímo konkrétní skupinu nebo se může opakovat stejný postup jako při prvním přiřazení práv skupině.



OBRÁZEK 13 - PŘIŘAZENÍ PRÁV SKUPINĚ DEFINOVANÉ V PLUGINU VIRTUALGROUP

5.3. EFEKTIVITA SPRÁVY UŽIVATELSKÝCH PRÁV PRO STUDENTSKÉ PROJEKTY

Celá koncepce správy práv je v DokuWiki postavena na jednoduchém modelu ACL. Kde je definováno sedm úrovní oprávnění.

Název	Úroveň	Lze aplikovat na	Práva
admin	255	administrátorské pluginy	superuživatel
mazání	16	jmenné prostory	možnost nahraná data přepsat/smazat
upload	8	jmenné prostory	možnost nahrát data
vytváření	4	jmenné prostory	možnost vytvářet nové stránky
editace	2	stránky a jmenné prostory	možnost editovat existující stránky
čtení	1	stránky a jmenné prostory	čtení
žádné	0	stránky a jmenné prostory	žádný přístup

TABULKA 7 - PRÁVA JAK JSOU DEFINOVÁNA PRO DOKUWIKI

Výchozí skupiny

- Skupina „ALL“ označuje všechny nepřihlášené (anonymní) uživatele. Implementovaný web umožňuje této skupině zobrazit pouze obsah úvodní stránky.
- Skupina „user“ jejími členy jsou automaticky všichni úspěšně autentizovaní uživatelé.

Nové skupiny

- Skupina „skAdmin“ je v systému začleněna jako výchozí administrátorská skupina. Její členové jsou automaticky administrátory celé DokuWiki
- Skupina „LIPTEL“ umožňuje členům plný přístup k jmennému prostoru kat440:liptel

PŘÍKLAD APLIKACE PRÁV

Jako příklad zde popíši aplikaci práv na jeden konkrétní projekt. Práva budou definována pro skupinu „skTest01“ a pro hierarchii jmenných prostorů uvedenou níže.

kat440:studentske_projekty:2011_2012:letni:uvod_do_komunikacnich_technologii:projekt01

Práva pro - studentske_projekty

Práva jsou definována tak, aby jakýkoliv přihlášený uživatel mohl procházet strukturou projektů, pokud není v podřízeném jmenném prostoru stanoveno jinak.

Stránka/Jmenný prostor	Uživatel/Skupina	Oprávnění	Vymazat
*	@skAdmin	○ Žádné ○ Čtení ○ Úpravy ○ Vytvoření ○ Upload ◉ Mazání	☐
*	@ALL	○ Žádné ◉ Čtení ○ Úpravy ○ Vytvoření ○ Upload ○ Mazání	☐
*	@user	○ Žádné ◉ Čtení ○ Úpravy ○ Vytvoření ○ Upload ○ Mazání	☐
kat440:*	@ALL	◉ Žádné ○ Čtení ○ Úpravy ○ Vytvoření ○ Upload ○ Mazání	☐
kat440:studentske_projekty:*	@user	○ Žádné ◉ Čtení ○ Úpravy ○ Vytvoření ○ Upload ○ Mazání	☐

TABULKA 8 - NASTAVENÍ PRÁV PRO JMENNÝ PROSTOR "STUDENTSKE_PROJEKTY"

Práva pro - projekt01

U konkrétního projektu jsou práva definovaná tak, že k němu mají úplný přístup pouze členové skupiny administrátorů „skAdmin“ a členové skupiny „skTest01“. Zároveň ale členové skupiny „skTest“ nemohou modifikovat dokument „clenove“ a „zadani“.

Stránka/Jmenný prostor	Uživatel/Skupina	Oprávnění	Vymazat
*	@skAdmin	Žádné Čtení Úpravy Vytvoření Upload Mazání	
*	@ALL	Žádné Čtení Úpravy Vytvoření Upload Mazání	
*	@user	Žádné Čtení Úpravy Vytvoření Upload Mazání	
kat440:*	@ALL	Žádné Čtení Úpravy Vytvoření Upload Mazání	
kat440:studentske_projekty:*	@user	Žádné Čtení Úpravy Vytvoření Upload Mazání	
kat440:studentske_projekty:2011_2012:letni:uvod_do_komunikacnich_technologii:projekt01:*	@skTest01	Žádné Čtení Úpravy Vytvoření Upload Mazání	
kat440:studentske_projekty:2011_2012:letni:uvod_do_komunikacnich_technologii:projekt01:*	@user	Žádné Čtení Úpravy Vytvoření Upload Mazání	
kat440:studentske_projekty:2011_2012:letni:uvod_do_komunikacnich_technologii:projekt01:clenove	@skTest01	Žádné Čtení Úpravy Vytvoření Upload Mazání	
kat440:studentske_projekty:2011_2012:letni:uvod_do_komunikacnich_technologii:projekt01:zadani	@skTest01	Žádné Čtení Úpravy Vytvoření Upload Mazání	

TABULKA 9 - NASTAVENÍ PRÁV PRO JMENNÝ PROSTOR "PROJEKT01" A DOKUMENTY

6. PRAKTICKÁ REALIZACE NAVRŽENÉ DOKUWIKI

Od počátku je server provozován ve virtuálním prostředí. Katedra telekomunikační techniky disponuje virtualizačním prostředím na platformě VMware ESXi, na kterou byl server nasazen.

Zdroje alokované pro virtualizovaný server		**Virtualizovaný operační systém**	
CPU	Dvě procesorová jádra	**OS**	Debian 6.0.4 ("squeeze") [amd64]
RAM	2GB	**Typ Instalace**	web-server bez GUI
HDD	16GB	**Hostname**	kat440wiki
		IP	158.196.244.235

TABULKA 10 - INFORMACE O SERVERU

Aktualizace:

- Apache web server 2.2.16
- PHP 5.3.10-1

Instalace:

- php5-ldap
- exlive-latex-base
- imagemagick
- ghostscript
- freetype font library
- ntp deamon

6.1. INSTALACE DOKUWIKI

Pro instalaci byla použita verze DokuWiki 2010-11-07a „Anteater“ z oficiálních stránek projektu. Instalace byla provedena do „/var/www/wiki“.

Cesta	**Práva**		**Vlastník**	
	adresář	soubory	owner	group
bin	0770	0660	www-data	www-data
conf	0770	0660	www-data	www-data
data	0775	0664	www-data	www-data
inc	0770	0660	www-data	www-data
lib	0755	0644	www-data	www-data

TABULKA 11 - NASTAVENÍ PRÁV NA ADRESÁŘICH

Web server

V adresáři „/etc/apache2/sites-available/“ byly vytvořeny konfigurační soubory pro definici virtuálního hosta kat440wiki a kat440-ssl [Příloha C], [Příloha CD/DVD]. Pomocí příkazu “a2ensite” byly oba povoleny. Následně byl na základě vlastní autority vygenerován SSL certifikát pro HTTPS přístup (na podkladě IP adresy 158.196.244.235 – protože se jedná o testovací provoz nebyl zřízen DNS záznam). Certifikát byl uložen v „/etc/apache2/ssl-cert/kat440wiki-key-cert.pem“ (kořenový certifikát byl posléze umístěn na titulní stránku kat440wiki - https://158.196.244.235). Aby byl web dostupný pouze přes HTTPS, byl povolen „rewrite“ pomocí „a2enmod rewrite“. Rewrite je silný nástroj pro manipulaci s URL a umožňuje jejich změnu bez ohledu na požadavek uživatele. Pro tento případ je nastaveno pravidlo, které jakýkoliv požadavek na nezabezpečený web HTTP přemění na požadavek na HTTPS. Zároveň je tento mód využit pro funkci „pěkné url“, která je podporována přímo v DokuWiki.

Původní URL	http://example.net/doku.php?id=wiki:syntax
Pěkné URL	http://example.net/wiki/syntax

TABULKA 12 - ROZDÍLY V URL PŘI POUŽITÍ MÓDU "REWRITE"

Pro povolení použíti funkce „rewrite“ v DokuWiki byla nutná změna v konfiguračním souboru „/var/www/wiki/conf/local.php“ přidáním parametrů „$conf['userewrite']=1“ a „$conf['useslash']=1“ (nahrazuje dvojtečku lomítkem). Následně byl modifikován soubor „/var/www/wiki/.htaccess“ [Příloha D], [Příloha CD/DVD]. Nakonec byl proveden restart webového serveru.

Template

Jako výchozí grafický template byl použit projekt „Monobook“ [monobook], který přebírá rozvržení ovládacích prvku na základě MediaWiki. Template je umístěn v adresáři „/var/www/wiki/lib/tpl/monobook/“. Protože nebyl tento template zatím přeložen do češtiny, provedl jsem překlad, ten je umístěn v „/var/www/wiki/lib/tpl/monobook/lang/cs/“.

6.2. ZÁLOHOVÁNÍ

Zálohovací skript je umístěn v „/var/dokuwiki-backup/“ jeho název je „dw-backup-script“. Pro pravidelné spouštění skriptu je použit deamon „cron“. Akce je založena v „/etc/crontab“ a její provádění je naplánováno na každy den v 3:00.

```
0  3   * * *  root   /var/dokuwiki-backup/dw-backup-script
```

Jako externí uložiště pro zálohy mi byl poskytnut prostor na síťovém uložišti ve správě katedry. Pro přístup je použit protokol NFS. Toto uložiště je automaticky napojeno v „/mnt/nfs-backup/“. Pro automatické připojení při startu je použit „/etc/fstab“.

```
158.196.244.190:/dokuwiki-backup /mnt/nfs-backup
nfs rw,rsize=4096,wsize=4096,hard,intr,async,nodev,nosuid 0 0
```

7. ZÁVĚR

Předmětem této bakalářské práce byl návrh a implementace systému pro potřeby vedení dokumentace studentských semestrálních projektů.

Jedním z cílů bylo dosažení toho, aby se systém DokuWiki podařilo implementovat tak, že k autentizaci bude využíván centrální adresář VŠB-TU. Bylo proto nutné správně definovat konfiguraci serveru DokuWiki v návaznosti na službu adresáře a analyzovat a následně testovat tuto konfiguraci, aby autentizace probíhala korektně a spolehlivě. Tento krok potřebný pro dosažení tohoto cíle se podařilo i přes značnou pracnost nakonec zvládnout. Nebylo mým záměrem v této práci hovořit o administrativní náročnosti při zjišťování potřebných informací o struktuře, korektním využívání služby a získávání testovacích účtů, přesto však považuji za nutné se o tomto zmínit, neboť jsem musel oslovit přímo vedoucího oddělení CIT-IS, který mi věnoval svůj čas. Právě to, že jsem byl mnohdy závislý na aktivním přístupu někoho jiného, i když vstřícném, mi moji práci znesnadňovalo. Musím připustit, že na počátku se mi právě tyto komplikace nejevily jako podstatné, ale později, hlavně ve fázi testování, jsem musel vynaložit mnohem více času, než jsem původně předpokládal.

V návaznosti na autentizaci byl úspěšný i proces autorizace v samotném systému DokuWiki, tedy přiřazování práv uživatelům na základě loginu, který mají v adresáři. Následně pak s využitím pluginu „Virtual group“ je také možné definovat uživatelské skupiny a modifikovat jejich členy na základě loginu. Takto definovaným skupinám je následně možné v DokuWiki přiřazovat práva.

Celý obsah DokuWiki se podařilo v pravidelných intervalech zálohovat na externí datové uložiště pomocí skriptu, který prostřednictvím emailových zpráv informuje správce o úspěšném provedení. Zálohovací skript rovněž umožňuje řídit historii záloh a jejich rozsah.

Aby byl ušetřen čas při zakládání nových projektů na základě struktury popsané v kapitole 4.7, bylo zapotřebí vytvořit zcela nový plugin, který umožňuje jednoduše tuto strukturu generovat. Plugin s touto funkčností jsem úspěšně naprogramoval. Následně jsem se snažil tento plugin rozšířit o funkci automatického přiřazování práv skupinám uživatelů ke konkrétním projektům. Musím konstatovat, že se nepodařilo dořešit proces automatického generování skupiny pomocí propojení s pluginem „Virtual group“ na základě definice členství uživatelů přímo v konkrétních projektových jmenných prostorech. Jednou z příčin nefunkčnosti propojení je dle mého názoru i narušování mechanismů řízení přístupových práv a tím narušení celkové bezpečnosti DokuWiki. Proces přiřazování práv projektovým jmenným prostorům je tedy třeba i nadále řešit ručně, což je v procesu efektivní správy značným problémem. Na základě této skutečnosti a ve spojení s celkovou koncepcí systému DokuWiki je efektivní řízení uživatelských přístupů ze strany správce při dané hierarchii velmi náročné. Vzhledem k množství projektových jmenných prostorů a jednotlivých unikátních skupin řešící přístup členů k daným projektům je tedy navrhovaný systém neefektivní.

Hlavní přínos své práce spatřuji v úspěšném vyřešení otázky jak autentizovat uživatele prostřednictvím adresářové služby VŠB-TU do systému DokuWiki a v přímé návaznosti pak vyřešení otázky autorizace nejen uživatelů, ale i celých uživatelských skupin uvnitř systému DokuWiki. Dále pak vyřešení otázky, jak eliminovat opakované zakládání projektových hierarchii pro jednotlivé akademické roky, jejich semestry a jednotlivé předměty.

Závěrem tedy konstatuji, že je celý navržený systém plně funkční, ale pro nasazení v praxi se v současném stavu nejeví jako vhodný. Pokud by se podařilo efektivně generovat i skupiny a následně provádět automatické přiřazování práv, mohl by být projekt úspěšně nasazen a splnit tak požadované cíle.

SEZNAM POUŽITÉ LITERATURY

[1] KOSEK, Jiří. *PHP a XML*. 1. vyd. Praha: Grada, 2009, 367 s. ISBN 978-80-247-1116-4.

[2] PROCHÁZKA, David. *CSS a XHTML*: *tvorba dokonalých WWW stránek krok za krokem*. 2., aktualiz. vyd. Praha: Grada, 2011, 175 s. Průvodce (Grada). ISBN 978-80-247-3897-0.

[3] ODELL, Den. *JavaScript*: *průvodce programováním ajaxových aplikací*. Vyd. 1. Brno: Computer Press, 2010, 368 s. ISBN 978-80-251-2733-9.

[4] GILMORE, Jason W. *Velká kniha PHP a MySQL 5*: *kompendium znalostí pro začátečníky i profesionály*. Vyd. 1. [i.e. 2. vyd.]. Brno: Zoner Press, 2007, 864 s. ISBN 80-868-1553-6.

[5] *Historie PHP* [online]. 2012 [cit. 2012-04-27]. Dostupné z: http://php.net/manual/en/history.php

[6] Learn About Java Technology. *Http://www.java.com* [online]. 2012 [cit. 2012-04-27]. Dostupné z: http://www.java.com/en/about/

[7] PECINOVSKÝ, Rudolf. *Myslíme objektově v jazyku Java*: *kompletní učebnice pro začátečníky*. 2., aktualiz. a rozš. vyd. Praha: Grada, 2009, 570 s. ISBN 978-80-247-2653-3.

[8] DARWIN, Ian F. *Java*: *kuchařka programátora*. Vyd. 1. Brno: Computer Press, 2006, 798 s. ISBN 80-251-0944-5.

[9] CLARKE, Roger. Identification and Authentication Glossary [online]. 2004 [cit. 2008-12-15]. Dostupný z WWW: <http://www.anu.edu.au/people/Roger.Clarke/EC/IdAuthGloss.html>.

[10] STANEK, William R. *Active Directory*: *kapesní rádce administrátora*. Vyd. 1. Brno: Computer Press, 2009, 352 s. ISBN 978-80-251-2555-7.

[11] LOSHIN, Peter. *Big book of lightweight directory access protocol (LDAP) RFCs*. San Francisco, Calif.: Morgan Kaufmann, 2000. ISBN 01-245-5843-7.

[12] HASSELL, Jonathan. *Radius*. Vyd. 1. New York: O´Reilly, 2003, 190 s. ISBN 05-960-0322-6.

[13] What's Shibboleth?. SHIBBOLETH CONSORTIUM. *Shibboleth Consortium* [online]. 2012 [cit. 2012-04-27]. Dostupné z: http://www.shibboleth.net/about/index.html

[14] Authentication Backends. *Dokuwiki.org* [online]. 2012 [cit. 2012-04-27]. Dostupné z: http://www.dokuwiki.org/auth

[15] Backup data from dokuwiki. *Http://www.dokuwiki.org* [online]. 2011, 2011 [cit. 2012-04-27]. Dostupné z: http://www.dokuwiki.org/tips:backup_script

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

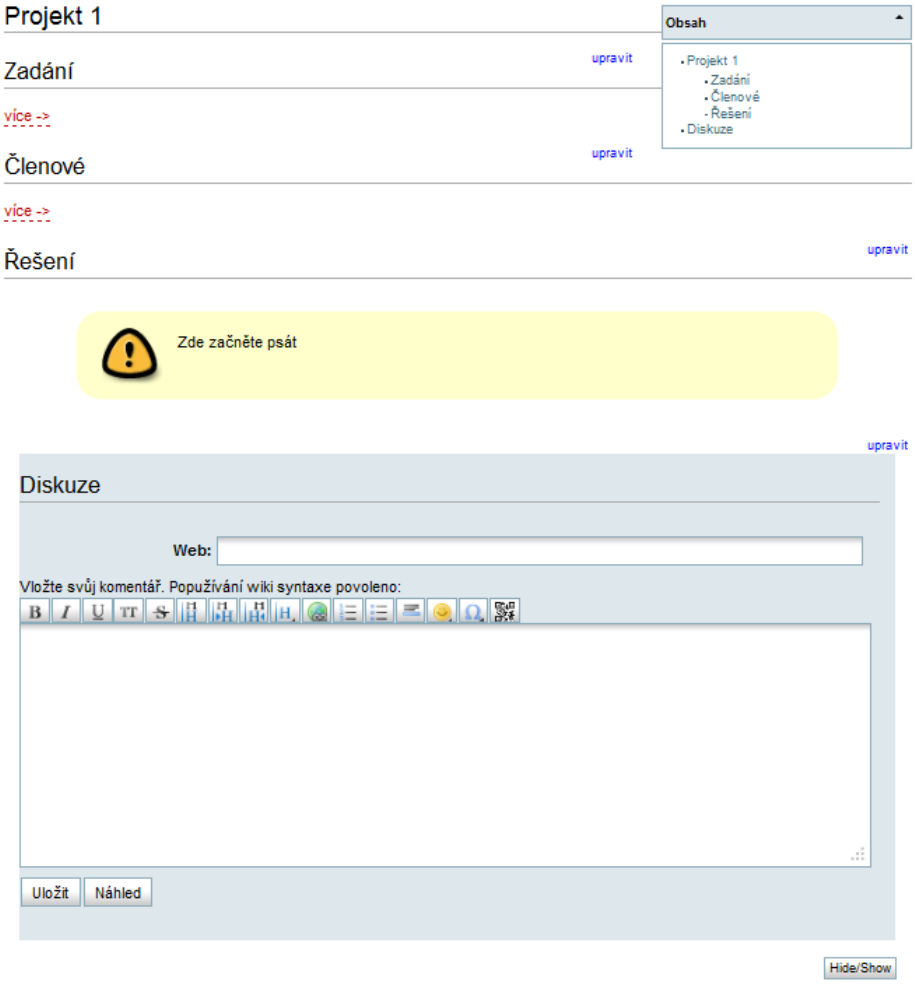
PŘÍLOHY

PŘÍLOHA NA CD/DVD

- **apache** – *definice sítě a konfigurace PHP*
  - **sites-available**
    - kat440wiki
    - kat440wiki-ssl
  - php.ini
- **dokuwiki-backup-script** – *zálohovací skript*
  - dw-backup-script
- **dokuwiki-plugin** – *plugin pro generování hierarchií*
  - **hierarchyconfig**
    - **conf**
      - metadata.php
    - **images**
      - user_delete.png
      - user_edit.png
    - **lang**
      - **cs**
        - lang.php
        - settings.php
      - **en**
        - lang.php
        - settings.php
    - Admin.php
    - manager.dat
    - plugin.info.txt
    - style.css
- **kat440wiki-complet** – *kompletní archiv celé DokuWiki*
  - dokuwiki-kat440.tar.bz2
- **monobook-template** – *překlad templatu Monobook*
  - **lang**
    - **cs**
      - .htaccess
      - index.html
      - lang.php
      - settings.php
      - style.css
- content.txt – *obsah*
- crc.sha – *kontrolní součet*

PŘÍLOHA A – OBSAH PROJEKTOVÉ STRÁNKY A JEJÍ VÝSLEDNÝ VZHLED

```
====== {project_title} ======

===== Zadání =====

{{page>zadani&firstseconly&nofooter&noeditbtn&noheader}}

[[zadani|více ->]]

===== Členové =====

{{page>clenove&firstseconly&nofooter&noeditbtn&noheader}}

[[clenove|více ->]]

===== Řešení =====

<note important>Zde začněte psát</note>

----

~~DISCUSSION~~
```



PŘÍLOHA B – ZÁLOHOVACÍ SKRIPT

```
#!/bin/bash
# DokuWiki Backup Script
# kri003@vsb.cz
# kat440wiki backup - based on dw-backup.sh 328 2004-12-22 13:15:20Z dp $
# v 1.1 - 20.3.2012 18:31:26
# script start time and date
STARTTIME="`date '+%s'`"
DATE="`date '+%d/%m/%y %H:%M:%S'`"
## config
# path to wiki (nosymlink allowed!)
WIKIPATH="/var/www/wiki"
# backup name data-${BACKUPNAME} / media-${BACKUPNAME} / wiki-${BACKUPNAME}
BACKUPNAME="$(hostname)"
# backup path
BACKUPPATH="/mnt/nfs-backup/kat440-dokuwiki-backup"
# count of backups
DAILY_DATA_BACKUPS="14" # DATA
DAILY_MEDIA_BACKUPS="14" # MEDIA
DAILY_CONF_BACKUPS="14" # backup WIKI without data
# comma separated email address for info mail
INFOMAIL="kri003@vsb.cz"
## no more config
# creates $1, if not existant
checkDir()
{
if [ ! -d "${BACKUPPATH}/$1" ]
then
mkdir -p "${BACKUPPATH}/$1"
fi
}
# 1 -> path
# 2 -> name
# 3 -> number of backups
rotateDir()
{
for i in `seq $(($3 - 1)) -1 1`
do
if [ -f "$1/$2.$i.tar.bz2" ]
then
mv "$1/$2.$i.tar.bz2" "$1/$2.$((i + 1)).tar.bz2"
fi
done
}
# make sure everything exists
checkDir "data"
checkDir "data/archive"
checkDir "data/daily"
checkDir "media"
checkDir "media/archive"
checkDir "media/daily"
checkDir "wiki"
checkDir "wiki/archive"
checkDir "wiki/daily"
# first step: rotate daily.
rotateDir "${BACKUPPATH}/data/daily" "data-${BACKUPNAME}" "$DAILY_DATA_BACKUPS"
rotateDir "${BACKUPPATH}/media/daily" "media-${BACKUPNAME}" "$DAILY_MEDIA_BACKUPS"
rotateDir "${BACKUPPATH}/wiki/daily" "wiki-${BACKUPNAME}" "$DAILY_CONF_BACKUPS"
# then create our backup
nice -n 10 tar --exclude="data/cache/*" --exclude=".*" --exclude="data/media" -cjf
"${BACKUPPATH}/data/data.1.tar.bz2" -C "${WIKIPATH}" "data"
nice -n 10 tar --exclude=".*" -cjf "${BACKUPPATH}/media/media.1.tar.bz2" -C "${WIKIPATH}" "data/media"
nice -n 10 tar -cjf "${BACKUPPATH}/wiki/wiki.1.tar.bz2" "${WIKIPATH}" --exclude="data" -P
# create an archive backup? Day in month.
if [ `date +%d` == "01" ]
then
cp "${BACKUPPATH}/data/data.1.tar.bz2" "${BACKUPPATH}/data/archive/archive-data-${BACKUPNAME}-"`date
+%m-%d-%Y`".tar.bz2"
cp "${BACKUPPATH}/media/media.1.tar.bz2" "${BACKUPPATH}/media/archive/archive-media-${BACKUPNAME}-
"`date +%m-%d-%Y`".tar.bz2"
cp "${BACKUPPATH}/wiki/wiki.1.tar.bz2" "${BACKUPPATH}/wiki/archive/archive-wiki-${BACKUPNAME}-"`date
+%m-%d-%Y`".tar.bz2"
fi
```

```
# add them to daily.
DATABACKUP="${BACKUPPATH}/data/daily/data-${BACKUPNAME}.1.tar.bz2"
MEDIABACKUP="${BACKUPPATH}/media/daily/media-${BACKUPNAME}.1.tar.bz2"
WIKIBACKUP="${BACKUPPATH}/wiki/daily/wiki-${BACKUPNAME}.1.tar.bz2"
mv "${BACKUPPATH}/data/data.1.tar.bz2" "${DATABACKUP}"
mv "${BACKUPPATH}/media/media.1.tar.bz2" "${MEDIABACKUP}"
mv "${BACKUPPATH}/wiki/wiki.1.tar.bz2" "${WIKIBACKUP}"
# enum size of backups
FILESIZEDATA=$(du -h "${DATABACKUP}")
FILESIZEMEDIA=$(du -h "${MEDIABACKUP}")
FILESIZEWIKI=$(du -h "${WIKIBACKUP}")
# store end time
ENDTIME="`date '+%s'`"
# store run time - tics
TIME=$(echo "${ENDTIME} - ${STARTTIME}"|bc )
# transforme TIME to h:m:s
((h=${TIME}/3600))
((m=${TIME}%3600/60))
((s=${TIME}%60))
# store file backup run time
RUNTIME=$(printf "%dh:%dm:%ds\n" $h $m $s)
# if info mail declared
if [ ${INFOMAIL} ]
then
sendMail()
{
SUBJECT="[BACKUP] $(hostname) ${DATE}"
BODY="Backup Info
Runtime: ${RUNTIME}
Allocated size
${FILESIZEDATA}
${FILESIZEMEDIA}
${FILESIZEWIKI}
"
echo "${BODY}" > /tmp/dokuwiki-backup-mail # temp mail body
mail -s "${SUBJECT}" "${INFOMAIL}" < /tmp/dokuwiki-backup-mail # read body from temp file
}
sendMail # send message
fi
## EOF
```

PŘÍLOHA C – HOST KAT440WIKI A KAT440-SSL

kat440wiki

```
<VirtualHost *:80>
        ServerAdmin webmaster@kat440wiki
        ServerName 158.196.244.235
        ServerAlias kat440wiki.local
        DocumentRoot /var/www
        DirectoryIndex wiki/index.php
        <Directory />
                Options FollowSymLinks
                AllowOverride All
        </Directory>
        <Directory /var/www/>
                php_admin_value open_basedir "/var/www/wiki/:/tmp/"
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

kat440wiki-ssl

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
        ServerAdmin webmaster@kat440wiki
        ServerName 158.196.244.235
        ServerAlias kat440wiki.local
        DocumentRoot /var/www
        DirectoryIndex wiki/index.php
        <Directory />
                Options FollowSymLinks
                AllowOverride All
        </Directory>
        <Directory /var/www/>
                #php_admin_value open_basedir "/var/www/wiki/:/tmp/"
                #Options Indexes FollowSymLinks MultiViews
                #AllowOverride All
                #Order allow,deny
                #allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log

        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined

        SSLEngine on

        SSLCertificateFile    /etc/apache2/ssl-cert/kat440wiki-key-cert.pem

        #<FilesMatch "\.(cgi|shtml|phtml|php)$">
        #       SSLOptions +StdEnvVars
        #</FilesMatch>
```

```
        #<Directory /usr/lib/cgi-bin>
        #       SSLOptions +StdEnvVars
        #</Directory>

        BrowserMatch "MSIE [2-6]" \
                nokeepalive ssl-unclean-shutdown \
                downgrade-1.0 force-response-1.0
        # MSIE 7 and newer should be able to use keepalive
        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
</IfModule>
```

PŘÍLOHA D – DOKUWIKI .HTACCESS

```
## Enable this to restrict editing to logged in users only

## You should disable Indexes and MultiViews either here or in the
## global config. Symlinks maybe needed for URL rewriting.
Options -Indexes -MultiViews +FollowSymLinks

## make sure nobody gets the htaccess, README, COPYING or VERSION files
<Files ~ "^([\._]ht|README$|VERSION$|COPYING$)">
    Order allow,deny
    Deny from all
    Satisfy All
</Files>

## Uncomment these rules if you want to have nice URLs using
## $conf['userewrite'] = 1 - not needed for rewrite mode 2
RewriteEngine on

RewriteBase /wiki

RewriteCond %{HTTPS} !=on
RewriteRule .* https://%{HTTP_HOST}%{REQUEST_URI} [R,L]

RewriteRule ^_media/(.*)              lib/exe/fetch.php?media=$1  [QSA,L]
RewriteRule ^_detail/(.*)             lib/exe/detail.php?media=$1  [QSA,L]
RewriteRule ^_export/([^/]+)/(.*)     doku.php?do=export_$1&id=$2  [QSA,L]
RewriteRule ^$                        doku.php  [L]
RewriteCond %{REQUEST_FILENAME}       !-f
RewriteCond %{REQUEST_FILENAME}       !-d
RewriteRule (.*)                      doku.php?id=$1  [QSA,L]
RewriteRule ^index.php$               doku.php
```