Univerzita Pardubice

Fakulta Elektrotechniky a Informatiky

Protokol LDAP a možnosti jeho využití

Jaroslav Šafář

Bakalářská práce

2010

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Jaroslav ŠAFÁŘ**
Osobní číslo:	**I07968**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Protokol LDAP a možnosti jeho využití**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování:

Teoretická část
Popis protokolu LDAP - Lightweight Directory Access Protocol. Návrh modelu využití LDAP s MySQL databází v informačním systému.
Implementační část
Realizace návrhu a implementace modelu pro informační systém. Vytvoření jednoduché webové aplikace využívající navrhnutý model.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

*GILMORE, W. Jason. Velká kniha PHP a MySQL 5 : kompendium znalostí pro začátečníky i profesionály. RNDr. *Jan Pokorný. [s.l.] : [s.n.], 2007. 864 s. ISBN 80-86815-53-6.

*BENÁK, Karel. Použití adresářových služeb v informačních systémech. [s.l.], 2004. 54 s. Diplomová práce. *Dostupný z WWW: <http://ldap.benak.net/diplom.pdf>.

*OpenLDAP [online]. 2009 [cit. 2010-01-10]. Dostupný z WWW: <http://www.openldap.org/>.

*BURDA, Zdeněk. Využití LDAPu v praxi [online]. 2005 [cit. 2010-01-10]. Dostupný z WWW: <http://ldap.zdenda.com/>.

Vedoucí bakalářské práce: **Ing. Martin SEMERÁD**
eBrana

Datum zadání bakalářské práce: **15. ledna 2010**

Termín odevzdání bakalářské práce: **14. května 2010**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2010

Prohlášení autora

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 14. 5. 2010

Jaroslav Šafář

Poděkování

Děkuji svému vedoucímu ing. Martinu Semerádovi za cenné podněty, náměty a připomínky.

Anotace

Cílem této práce je popis adresářového protokolu LDAP - Lightweight Directory Access Protocol a možnosti využití adresářové služby LDAP jako součásti informačního systému. Tato práce popisuje adresářový protokol LDAP. Dále popisuje zprovoznění LDAP serveru a jeho konfiguraci pomocí softwaru OpenLDAP, který umožňuje spojení s relačními databázemi SQL.

Klíčová slova

adresářové služby, LDAP, server, autentizace, databáze, PHP

Title

Protocol LDAP and its utilization possibilities

Abstrakt

The aim of my graduation thesis is a description of directory protocol LDAP - the Lightweight Directory Access Protocol – and possibilities of directory service LDAP utilization as a part of an information system. This thesis describes the directory protocol LDAP. Next it describes an installation of LDAP server and its configuration through OpenLDAP software which enables connection with relational database SQL.

Key words

directroy services, LDAP, server, autentization, databases, PHP

Obsah

Seznam obrázků

Seznam zkratek

LDAP	Lightweight Directory Access Protocol
SQL	Structured Query Language
PHP	Hypertext Preprocesor

Úvod

Úkolem této práce je zprovoznění adresářové služby, konkrétně zprovoznění LDAP serveru, která bude využívána jako součást informačního systému ve společnosti eBRÁNA. Nová adresářová služba by měla sloužit jako centralizovaná databáze uživatelů, a to hlavně jako zdroj jejich přihlašovacích a identifikačních údajů. Znamená to tedy, že se pomocí této nové adresářové služby bude provádět autentizace uživatelů různých aplikací, které firma využívá. V této funkci by měla nahradit současnou databázi uživatelů, která má podobu databáze MySQL.

Požadavkem společnost eBRÁNA je zachování současné databáze uživatelů i s jejím obsahem, tak aby byl stále aktuální a použitelný. Proto je potřeba v další části zprovoznění LDAP serveru prozkoumat možnost, kde bude tuto podmínku možno splnit a nový adresářový server bude databázi MySQL využívat jako zdroj dat.

Této varianty lze dosáhnout pomocí softwaru OpenLDAP. Tato implementace LDAP umožňuje využít, jako své datové úložiště, relační databázi SQL. To znamená, že nová adresářová služba bude data čerpat z aktuální databáze uživatelů. Nemusí se tedy nic exportovat a jakékoli zásahy do uživatelských údajů, se mohou provádět stále pomocí aplikací pro správu MySQL databáze. Tato možnost backendu OpenLDAP na databázi SQL je v dostupných zdrojích uváděna jako experimentální, proto praktická část této práce otestuje toto spojení.

1 Adresářové služby

1.1 Adresářová služba

Adresářová služba je aplikace pracující s adresářem, sloužící k ukládání, organizaci a k zpřístupnění záznamů adresáře. Samotný adresář je organizační jednotka, která sdružuje záznamy různého charakteru tak, aby byla pro uživatele práce sním snadná, srozumitelná a adresář byl přehledný. Záznamy adresáře nebo adresářového serveru jsou tvořeny pomocí standardních schémat a každý záznam je složen z daných atributů, které definují daná schémata. Atributy záznamu jsou nositeli informací, které do adresáře vkládáme. Organizace záznamů v adresáři je hierarchická a má podobu stromové struktury. Příkladem jednoduchého adresáře může být telefonní seznam, kde máme záznam s atributem Jméno a s atributem Číslo a záznamy jsou řazeny podle abecedy. Dalo by se říci, že adresář je specializovaná databáze, ale rozhodně jej nelze zařadit do skupiny relačních databází, od kterých se liší vlastním použitím [1, 2, 3].

1.2 Adresářové služby vs. relační databáze

V porovnání s relačními databázemi jsou adresářové služby určeny především k vyhledávání. To znamená k rychlému a efektivnímu čtení záznamu adresáře. Neposkytují tedy oproti relačním databázím pokročilé databázové techniky jako jsou transakce, kontrola integrity dat nebo zadávání velmi složitých dotazů. Nejsou určeny ani pro data, která vyžadují časté zápisy nebo změny. Adresářové služby mají také sadu předefinovaných schémat, díky kterým není nutné definovat vlastní schéma uložení dat, jako je tomu u relačních databází. Předefinovaná schémata a jejich standardizace tak umožňuje lepší centralizaci dat a umožňují využití adresářové služby u více aplikací najednou [1, 2, 3].

1.3 Využití adresářových služeb

I přes určité nedostatky, o proti relačním databázím, mají adresářové služby široký okruh využití. Hojné využití adresářové služby se nachází v uchovávání informací o uživatelích(např. emailové účty, přihlašovací údaje atp.) [1]. Díky takto uchovaným informacím můžeme pomocí adresářové služby provádět jednoduchou autentizaci uživatele nebo ověřování uživatelský práv. Mnohem větší využití adresářové struktury může být například server DNS - Domain Name Systém [3].

2 Protokol LDAP

LDAP je zkratkou pro Lightweight Directory Access Protocol. Je to adresářový protokol pro ukládání a přístup k datům na adresářovém serveru [1]. Protokol LDAP je odvozen od adresářového protokolu X.500. Protokol X.500 je standardem ISO a uspořádává záznamy do hierarchického jmenného prostoru. Je však příliš náročný, a proto vznikl protokol LDAP jako jeho "odlehčená" verze. Protokol LDAP pracuje nad protokolem TCP/IP a výrazně zjednodušuje některé operace protokolu X.500 [3].

Protokol LDAP pracuje na bázi klient – server. Komunikace probíhá tak, že klient se připojí na příslušný adresářový LDAP server a zašle požadovaný dotaz. Klientem vytvořený dotaz obsahuje co nejpřesnější umístění záznamu v adresářovém stromě a dále pak může obsahovat upřesňující vyhledávací filtry a v případě potřeby autentizační údaje uživatele. Server na jeho dotaz odpoví příslušným výsledkem nebo sérií výsledků [1, 3]. Adresářový protokol LDAP je dále popsán a definován souborem čtyř modelů.

- Informační model;
- Jmenný model;
- Funkční model;
- Bezpečnostní model;

2.1 Informační model LDAP

Informační model LDAP nám definuje jaký datový typ a informace se bude na adresářový server zaznamenávat [1]. Záznam adresáře je soubor informací, kterými se snažíme popsat konkrétní objekt, například uživatele. Vytváření záznamů v adresáři je podrobně definováno v souboru schémat [2, 3].

Schémata adresářového serveru jsou souborem definic objektových tříd (*objectclass*), jejich atributů (*attributetype*), pravidel pro porovnávání atributů a syntaxí, které daná adresářová služba podporuje. Adresářový server zahrnuje již standardizovaná schémata, ve kterých jsou definovány typické podoby záznamů, například zmíněný uživatel (varianta schématu *inetOrgPerson*). V případě potřeby je možné vytvořit schéma vlastní. Je výhodnější se takové možnosti vyhnout, protože při tvorbě vlastních schémat by

mohlo dojít k nesnadné integraci aplikace. Zde je uveden příklad konkrétní definice objektové třídy ve schématu *inetOrgPerson.schema* [1, 2, 3].

```
objectclass   ( 2.16.840.1.113730.3.2.2
NAME 'inetOrgPerson'
    DESC 'RFC2798: Internet Organizational Person'
    SUP organizationalPerson
    STRUCTURAL
MAY (
    audio $ businessCategory $ carLicense $
    departmentNumber $ displayName $ employeeNumber $
    employeeType $ givenName $ homePhone $
    homePostalAddress $ initials $ jpegPhoto $
    labeledURI $ mail $ manager $ mobile $ o $ pager $
    photo $ roomNumber $ secretary $ uid $
    userCertificate $ x500uniqueIdentifier $
    preferredLanguage $ userSMIMECertificate $ userPKCS12 )
)
```

Každý záznam adresáře je tedy instancí objektové třídy definované ve schématu. Jednotlivé objektové třídy mohou mít za předka jinou objektovou třídu. Každá objektová třída slouží k popisu konkrétního objektu, který obsahuje souhrn atributů, které nám konkrétní objekt popisují [1].

Atributy objektové třídy nám slouží jako nositelé informací [1]. Atribut nese informaci, kterou je popsána jedna vlastnost objektu. Jejich definice se nachází společně s definicí objektových tříd ve schématu.

```
attributetype ( 2.16.840.1.113730.3.1.241
    NAME 'displayName'
    DESC 'RFC2798: preferred name to be used when
displaying entries'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
    SINGLE-VALUE )
```

2.1.1 LDIF

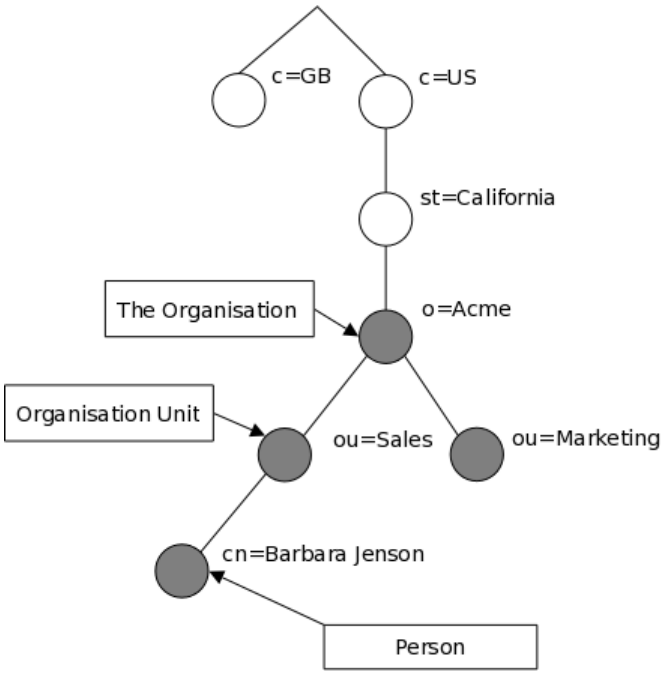
Nepatří do definice informačního modelu, ale je to formát, který slouží k reprezentaci záznamů LDAP, tak jej zmíním zde. LDIF je zkratkou pro LDAP Data Interchange Format. Je to standardizovaný formát pro reprezentaci a aktualizaci dat na adresářovém serveru. Jedná se o jednoduchou textovou reprezentaci záznamů v adresáři. Každý záznam je zde reprezentován svým rozlišujícím jménem, objektovou třídou, které je záznam instancí, a souborem atributů, které záznam obsahuje [9].

```
dn: cn=Frantisek Vonasek,dc=ebrana,dc=cz
objectclass: inetOrgPerson
cn: Frantisek Vonasek
sn: Vonasek
```

Využít formát LDIF lze například při importu a exportu dat z adresářového serveru. Lze jej použít i při aktualizačních operacích LDAP serveru, které jsou popsány v kapitole Funkční model.

2.2 Jmenný model LDAP

Jmenný model slouží k definici způsobu organizace záznamů v adresáři a způsobu, jakým se bude možné na daný záznam odkázat. Záznamy adresáře jsou ukládány do hierarchické stromové struktury, tzv. DIT, což je zkratka pro Directory Information Tree. Na Obrázku 1 můžeme vidět strukturu LDAP adresářového stromu [1, 2, 3].



Obrázek 1 – LDAP adresářový strom

zdroj: http://www.openldap.org/doc/admin24/intro.html

Každý záznam adresáře je, v rámci celého adresářového stromu, jednoznačně identifikovatelný svým rozlišovacím jménem (DN - Distinguished Name) [2]. Takové rozlišovací jméno je vlastně cesta k zpřístupnění daného záznamu. Podobu rozlišovacího jména, podle Obrázku 1, vidíme zde:

```
cn=Barbara Jenson,ou=Sales,o=Acme,st=Kalifornia,c=US
```

Celé rozlišovací jméno můžeme dále rozdělit. První část rozlišovacího jména, v případe obrázku je to atribut cn, je relativní rozlišovací jméno (RDN - Relative Distinguished Name). Relativní rozlišovací jméno nám udává jeden z atributů záznamu, podle kterého může být jednoznačně určený záznam, v rámci dané větve adresářového stromu. Takže v dané větvi se již nemůže nacházet záznam se stejným relativním rozlišovacím jménem.

Zbylá část rozlišovacího jména se skládá z části, která udává umístění záznamu v adresáři a části, která je pro všechny záznamy stejná. Tou je tzv. sufix. Sufix je kořen adresáře, který bývá odvozen od lokality umístění nebo od internetové domény [2]. Zaručuje jedinečnost adresáře.

V případě Obrázku 1 bude sufix:

```
o=Acme,st=Kalifornia,c=US
```

2.3 Funkční model LDAP

Funkční model v sobě obsahuje operace, kterými lze manipulovat se záznamy adresáře. Protokol LDAP má tři skupiny operací s adresářem [1, 2].

- Autentizační operace;
- Dotazovací operace;
- Aktualizační operace;

V jednotlivých implementacích LDAP je možné při práci s adresářovým serverem použít předem připravené nástroje. V popisu jednotlivých operací jsou využívány konzolové příkazy implementace OpenLDAP.

2.3.1 Autentizační operace

Jak název skupiny říká, jedná se o skupinu operací, která slouží k autentizaci uživatele při navázané komunikaci se serverem. U LDAP jsou to dvě autentizační operace *bind* a *unbind* [1, 3].

Operace *bind* slouží k autentizaci uživatele, který navázal komunikaci s adresářovým serverem. Podle výsledku autentizační operace a podle nastavených uživatelských přístupových práv je přihlášenému uživateli dovoleno pracovat s daty v adresáři pomocí dotazovacích a aktualizačních operací. Opakem je **operace *unbind***, která slouží k ohlášení uživatele, který navázal komunikaci se serverem.

Provádění dalších operací s daty adresářového serveru, hlavně pak u Aktualizačních operací, je závislé na nastavení přístupových práv uživatelů, které je blíže popsáno v kapitole Bezpečnostní model.

2.3.2 Dotazovací operace

Do této skupiny operací patří operace pro získávaní dat z LDAP adresáře. Patří sem operace pro vyhledávání záznamů adresáře a operace pro porovnávání dat obsažených v atributech záznamu adresáře.

Vyhledávací **operace *search*** se provede pomocí nástroje ***ldapsearch***. Nástroj *ldapsearch* má několik vstupních parametrů, kterými lze ovlivnit způsob a výsledek vyhledávání. Máme například možnost určit výchozí bod vyhledávání (*-b searchbase*), určit uživatele pro operaci bind (*-D binddn*), můžeme použít i vyhledávací filtry a seznam atributů, které mají být ve výsledku obsaženy [3]. U tohoto příkazu dojde k autentizaci uživatele *root*, kterému bude jako výsledek vyhledávání navrácen seznam zadaných atributů všech objektů adresáře.

```
ldapsearch -b "dc=ebrana,dc=cz" -D
"cn=root,dc=ebrana,dc=cz" –W objectclass="*" cn sn
```

Porovnávací **operace *compare***, slouží k porovnávání hodnoty atributu daného záznamu. Porovnávací operaci provedeme pomocí nástroje ***ldapcompare***. Základem je rozlišovací jméno záznamu, u kterého se porovnává hodnota atributu a hodnota daného atributu. V případě shody bude výsledek operace roven logické hodnotě *TRUE*. V opačném případě, bude výsledek *FALSE* [3].

```
ldapcompare "uid=vonasek,dc=ebrana,dc=cz"
userPassword:12345
```

2.3.3 Aktualizační operace

Do skupiny aktualizačních operací v LDAP patří operace pro přidání nového záznamu, editaci záznamu, přejmenování existujícího záznamu a mazání existujícího záznamu adresáře. Pro všechny aktualizační operace je většinou nezbytná autentizace uživatele, aby bylo možné ověřit práva, pro provádění těchto operací. Všechny operace lze provádět přímo v příkazové řádce nebo pomocí LDIF souborů. Použití LDIF souborů je vhodnější pro svou přehlednost a možnosti zpětné kontroly [1, 3].

Přidání nového záznamu do adresáře, **operace *add***, se provede pomocí nástroje ***ldapadd***. Vzhledem k nutnosti autentizace je jeden z parametrů autentizace uživatele. Pro zadní cesty k LDIF souboru přidáme parametr *–f název_souboru*.

```
ldapadd -D "cn=root,dc=ebrana,dc=cz" –W –f /soubor.ldif
```

LDIF soubor obsahuje rozlišovací jméno nového záznamu, atribut *changetype* a přidávané atributy záznamu. Atribut *changetype*, v LDIF souboru, slouží k určení plánované operace se záznamem.

```
dn: cn=Frantisek Vonasek,dc=ebrana,dc=cz
changetype: add
objectclass: inetOrgPerson
cn: Frantisek Vonasek
sn: Vonasek
```

U **operace *modify***, sloužící k modifikaci záznamu adresáře, má atribut *changetype* hodnotu *modify*. Modifikaci záznamu vyvoláme příkazem ***ldapmodify***, kde parametry jsou stejné jako u operace *ldapadd*. Rozdíly v LDIF souboru jsou v zápisu atributů. Atributy zde můžeme přidávat, měnit nebo mazat [3].

```
dn: cn=Frantisek Vonasek,dc=ebrana,dc=cz
changetype: modify
add: givenName
givenName: Frantisek
replace: cn
cn: Vonasek
delete: sn
sn: Vonasek
```

Další operace, **operace *modrdn***, slouží k přejmenování záznamu a lze ji použít i k přemístění záznamu adresáře. To znamená změnu rozlišovacího jména záznamu. Operace se provede pomocí nástroje ***ldapmodrdn***. Soubor LDIF obsahuje rozlišovací jméno záznamu a nové relativní rozlišovací jméno. Dále pak atributy s možností smazání starého záznamu a změny větve adresářového stromu [3].

```
dn: cn=Frantisek Vonasek,dc=ebrana,dc=cz
changetype: modrdn
newrdn: cn=Josef Vonasek
deleteoldrdn: 0
newsuperior: ou=People,dc=ebrana,dc=cz
```

Poslední operace je mazání záznamu, **operace *delete***. Provedeme ji příkazem ***ldapdelete***. Smazání záznamu proběhne po zadání rozlišovacího jména. Při použití LDIF souboru je nastaven atribut *changetype* na hodnotu *delete*. Detailnější popis všech

uvedených operací a seznamy parametrů, pro jejich provedení, najdete na manuálových stránkách projektu OpenLDAP[1].

2.4 Bezpečnostní model

2.4.1 Autentizace

Proces autentizace, tedy ověření identity uživatele, dochází v momentě, kdy se uživatel pokouší navázat spojení s LDAP serverem [1]. Proces autentizace může proběhnout několika způsoby.

Anonymní autentizace

Jedná se o způsob autentizace, kde se při autentizační operaci bind neposílají žádné uživatelské identifikační údaje. Při takové autentizaci uživatele je většinou přístup k datům velmi omezený [1].

Jednoduchá autentizace

V tomto případě jsou při operaci bind odeslány údaje o uživateli. Jedná se o uživatelovo rozlišovací jméno DN a jeho heslo, které je uloženo v atributu uživatelova záznamu [1].

Proxy autentizace

U této formy autentizace se využívá existence uživatele s právem nahlížet na uživatelská hesla. Tento uživatel má tedy moc porovnat zadané údaje s údaji v záznamu [1].

PKI autentizace

Zde se využívá digitálních PKI certifikátů. Každý uživatel má vlastní certifikát, který musí být totožný s certifikátem, který je uložen na serveru v atributu záznamu uživatele [1].

Mechanizmus SASL

SASL je zkratkou pro Simple Authentication Security Layer. Je to obecná metoda pro přidávání nebo zlepšování ověřování v protokolech klient/server. Mechanizmus SASL nabízí možnost rozhodnout se mezi několika mechanizmy pro ověřování uživatele.

[1] http://www.openldap.org/software/man.cgi

Zvolený ověřovací mechanizmus řídí komunikaci mezi klientem a serverem při ověřování uživatele [4].

2.4.2 TLS

Transport Layer Security je rozšíření komunikace TCP o šifrování pro zajištění soukromí a integrity zpráv. U protokolu LDAP je využito vrstev TLS pro ochranu komunikace. U jednoduché autentizace uživatele tak zabráníme, aby přihlašovací údaje uživatele nebyly odesílány v čistě textové podobě. Užitečná je i kombinace zabezpečení TLS s použitím mechanizmu SASL [4].

2.4.3 Autorizace

Proces autorizace následuje po úspěšném provedení operace autentizace. Úkolem autorizace je určení rozsahu oprávnění uživatele. Určuje rozsah přístupu k jednotlivým záznamům adresáře. Přístup k záznamů nám určují přístupová práva. Na příkladu zde je uvedeno nastavení přístupových práv serveru OpenLDAP.

Přístupová práva můžeme nastavit až do úrovně atributu záznamu [1, 2]. Nastavením přístupových práv můžeme například uživatelům zamezit přístup k jiným záznamům nebo atributům záznamu, než k jejich vlastním.

```
access to attrs="userPassword"
    by dn="cn=root,dc=ebrana,dc=cz" write
    by anonymous auth
    by self write
    by * none
```

Zde můžeme vidět nastavení přístupu ke konkrétnímu atributu, v tomto případě je to atribut nesoucí informaci o uživatelském heslu. Administrátor a vlastník záznamu má právo zápisu, a anonymní uživatelé mohou k tomuto atributu přistupovat pouze k potřebě autentizace.

2.5 Implementace LDAP serveru

Existuje celá řada implementací protokolu LDAP. Lze vybírat mezi open-source softwarem, kde jsou nejznámější produkty OpenLDAP a TinyLDAP a mezi komerčními produkty, mezi které patří Microsoft Active Directory. Jednotlivé implementace jsou rozdílné například ve způsobu konfigurace a způsobu ukládání dat [1].

2.5.1 Microsoft Active Directory

Active Directory je implementace adresářové služby LDAP. Jedná se o komerční produkt firmy Microsoft, který byl vyvinut pro použití v systému Windows. Nejedná se přímo o LDAP server. Služba Active Directory má mnoho součástí a je založena na mnoha technologiích. Protokol LDAP je zde využívám pro přístup k datům služby Active Directory. Zajišťuje tak komunikaci klientských počítačů se serverem [5].

2.5.2 TinyLDAP

TinyLDAP je velmi rychlá a velmi zjednodušená verze LDAP severu. Obsahuje pouze základní operace s LDAP serverem. Záznamy ukládá do textových souborů [7].

2.5.3 OpenLDAP

Software OpenLDAP je open-source[2] implementace adresářové služby, založená na protokolu LDAP, vyvíjena internetovou komunitou [3]. Je vyvíjena pro operační systémy Linux, FreeBSD, Solaris apod. Jedná se o nejrozšířenější implementaci adresářové služby, i když napracuje tak rychle jako jiné implementace, především ty komerční. Má nízkou hardwarovou náročnost. V základní verzi podporuje velké množství komunikačních mechanizmů a pro ukládání dat umožňuje využít velké množství databází [1].

[2] software s otevřeným zdrojovým kódem

3 LDAP a SQL databáze

Jedním z úkolů této práce je praktická zkouška spojení adresáře LDAP a relační databáze SQL. Konkrétně je úkolem využití relační databáze SQL jako zdroj dat pro LDAP adresář. V tomto případě, je k dispozici databáze MySQL, která slouží jako databáze uživatelů. Vytvořením serveru LDAP serveru chceme získat centralizovaný adresář uživatelů, pomocí kterého je možná autentizace uživatelů ve větším množství aplikací například přihlašování administrátorů webových stránek, přihlášení do vývojových aplikací (aplikace Mantis) apod.

Pro instalaci adresářové služby jsem vybral implementaci LDAP serveru OpenLDAP. Důvody k rozhodnutí použít pro zprovoznění adresářové služby implementaci OpenLDAP nám ukazuje SWOT analýza. SWOT analýza je znázorněna v Tabulce 1.

Tabulka 1 – SWOT analýza

Silné stránky:	**Slabé stránky:**
Open-source software; Nízká HW náročnost;	Nižší pracovní rychlost;
Příležitosti:	**Hrozby:**
Využití velkého množství komunikačních mechanizmů; Využití velkého množství databází hlavně pak využití databází SQL;	

Zdroj: Autor

Chceme-li používat databázi SQL jako databázi OpenLDAP serveru, je potřeba provést následující kroky [3]:

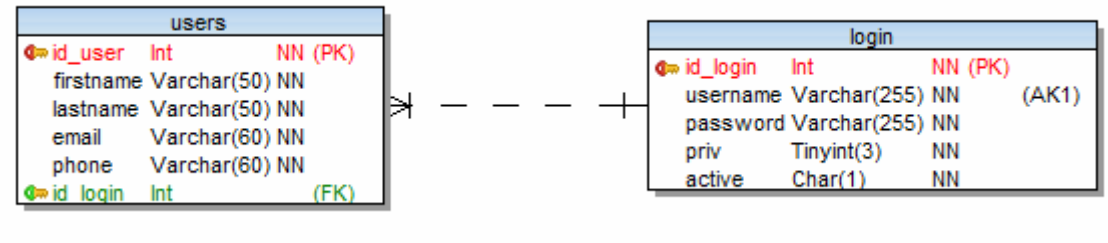
- Připravit databázi SQL (v našem případě MySQL);
- Nastavit mapování dat;
- Nastavení rozhraní ODBC;

3.1 MySQL databáze

MySQL je databázový systém využívající jazyka SQL. Je to multiplatformní databáze, která je velmi snadno implementovatelná. Je bezplatně k dispozici pod licencí

GPL, ale i pod komerční licencí. Pro svoje vlastnosti je velmi často používaná, například na webových serverech. V posledních verzích jsou již doplňovány chybějící funkce, jako jsou například pohledy, triggery a procedury [8].

Relační databáze MySQL v současnosti zastává, ve společnosti eBRÁNA, funkci adresáře, ve kterém jsou uloženy informace o uživatelích. Tato databáze nám bude sloužit jako zdroj dat pro náš LDAP server. Pro test je návrh databáze zjednodušen jen na potřebné tabulky, ve kterých jsou uloženy data o uživatelích. Ostatní tabulky pro nás nejsou důležité. Podoba zjednodušeného návrhu databáze je na Obrázku 2, kde je znázorněn E-R diagram.



Obrázek 2 – E-R diagram databáze uživatelů

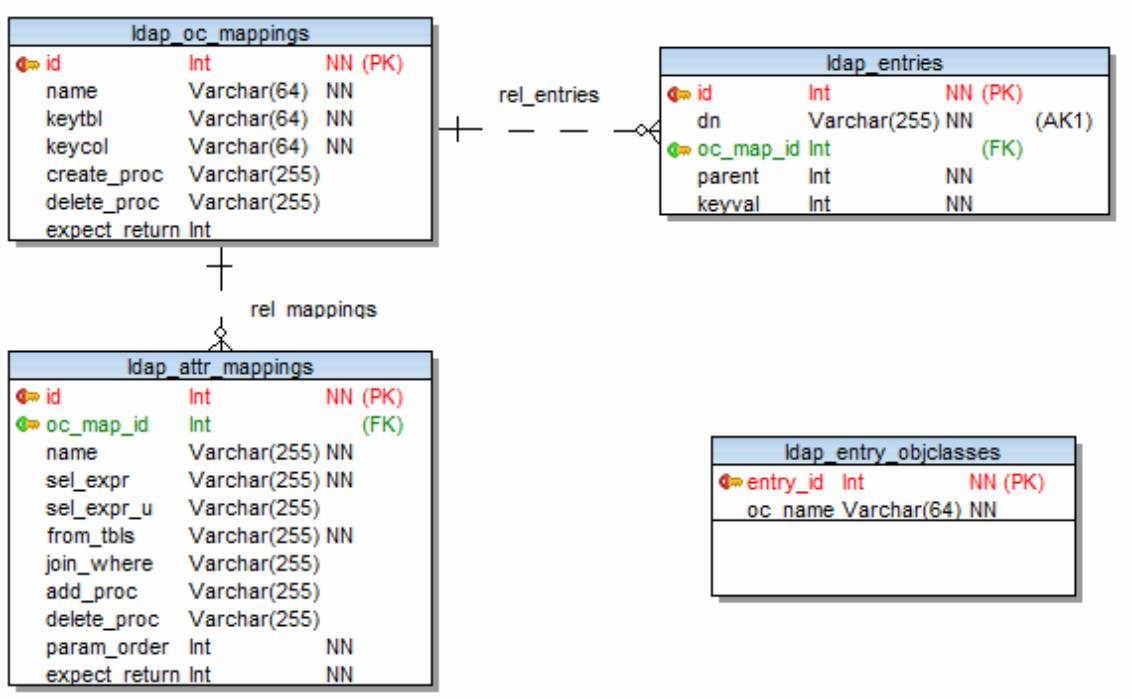
Zdroj: Autor

První tabulka *users* slouží k uchování informačních údajů o všech uživatelích. Jsou zde zaznamenávány údaje o uživateli jako je jméno a příjmení, telefon, email apod. Později je možné doplnit je o další informace. Důležitá je hodnota cizího klíče *id_login*. Ta nám udává hodnotu primárního klíče *id_login* v tabulce *login*. Určuje tedy vztah mezi oběma tabulkami, takže každému uživateli musí být přiřazen jeden záznam v tabulce *login*.

Tabulka *login*, pak obsahuje informace, které slouží k autentizaci uživatele v aplikaci, která daný návrh využívá. Jsou zde přihlašovací údaje, informace o úrovni privilegií a informace o aktivitě uživatele.

3.2 Nastavení mapování SQL databáze

Pro nastavení mapování dat SQL databáze je potřeba do současné databáze přidat mapovací tabulky LDAP. Podoba přidávaných mapovacích tabulek LDAP je vidět na Obrázku 3, kde je znázorněn E-R diagram.



Obrázek 3 – E-R diagram mapovacích tabulek

Zdroj: Autor

3.2.1 Popis mapovacích tabulek

Tabulky, zobrazené v E-R digramu na Obrázku 3, slouží k ukládání informací o mapování objektů, záznamů a jednotlivých atributů záznamu. Popíši funkce jednotlivých tabulek.

Tabulka ldap_oc_mappings

Tabulka ldap_oc_mappings slouží k mapování objektů (objektových tříd). Tabulka obsahuje sloupec *name*, osahující název objektu, sloupec *keytbl* s názvem tabulky, ke které se třída vztahuje, a sloupec *keycol*, kde je název sloupce vztažné tabulky, který je primárním klíčem. Zbylé sloupce jsou určeny pro názvy vytvořených procedur, které se vykonají v případě vytvoření (sloupec *create_proc*) nebo smazání (sloupec *delete_proc*) objektové třídy. Jejich deklarování je nepovinné [3].

id	name	keytbl	keycol	create_proc	delete_proc	expect_return
1	inetOrgPerson	users	id_users			0

Obrázek 4 – Tabulka ldap_oc_mappings

Zdroj: Autor

Na Obrázku 4 je zobrazen výpis tabulky. Je vidět definovaný objekt třídy inetOrgPerson, která se vztahuje k tabulce uživatelů, kde primárním klíčem, je sloupec id.

Tabulka ldap_attr_mappings

Touto tabulkou se provádí mapování jednotlivých atributů objektu. Každý atribut je zde určen příslušností k danému objektu, sloupcem *oc_map_id*, který má hodnotu id objektu, a názvem atributu (sloupec *name*). Mapování dat atributu se provádí pomocí sloupců *sel_expr*, *sel_expr_u*, *from_tbls* a *john_where*. Hodnotami těchto sloupců se vytvoří dotaz SELECT, pro získání hodnoty atributu. Povinné jsou sloupce s parametrem NOT NULL [3].

id	oc_map_id	name	sel_expr	sel_expr_u	from_tbls	join_where	add_proc	delete_proc	param_order	expect_return
1	1	uid	login.username	*NULL*	users,login	users.id_login=login.id_login	*NULL*	*NULL*	3	0

Obrázek 5 – Tabulka ldap_attr_mappings

Zdroj: Autor

Na Obrázku 5 je zobrazen řádek z tabulky obsahující atribut *uid*. Z hodnot sloupců určujících dotaz, získáme následující dotaz:

```
SELECT login.username FROM user,login
WHERE users.id_login=login.id_login;
```

Tabulka ldap_entries

Tabulka ldap_entries slouží k ukládání rozlišovacích jmen záznamů. Určuje podobu adresářového stromu. Objekt je určen sloupcem *oc_map_id*, podoba rozlišovacího jména záznamu je určena sloupcem *dn* a sloupec *parent* udává rodiče záznamu. Ten se určí pomocí id, nadřazeného záznamu v tabulce ldap_entries. Poslední sloupec *keyval*, obsahuje hodnotu id skutečného záznamu databáze [3].

id	dn	oc_map_id	parent	keyval
1	dc=ebrana,dc=cz	1	0	1
2	uid=nekdo,dc=ebrana,dc=cz	2	1	1
3	uid=nekdo2,dc=ebrana,dc=cz	2	1	2

Obrázek 6 – Tabulka ldap_entries

Zdroj: Autor

Tabulka ldap_entry_objclasses

Tato tabulka slouží jako pomocná v případech, kdy dochází k mapování záznamů, které zahrnují více objektových tříd. Při výpisu záznamu jsou použity i atributy jiných objektů [3].

3.2.2 ODBC

ODBC, neboli Open Database Connectivity, je standardizované rozhraní pro přístup k databázovým systémům. Úkolem ODBC je poskytovat přístup k datům, který je nezávislí na programovacím jazyku, operačním systému a databázovém systému [10]. Slouží jako mezičlánek pro spojení klientské aplikace a databázového serveru. V tomto případě je adresářová služba OpenLDAP klientská aplikace a databázový server je naše databáze MySQL.

3.2.3 Instalace a nastavení ODBC

Instalace rozhraní ODBC lze provést pomocí standardních balíků distribuce operačního systému linux. Pomocí příkazu *apt-get* nainstalujeme balík iodbc a balík libmyodbc, který obsahuje ovladače pro MySQL databázi. V našem případě bude nainstalováno rozhraní ODBC verze 3.52.6.

```
sudo apt-get install unixodbc libmyodbc
```

Po úspěšné instalaci nakonfigurujeme rozhraní ODBC pro přístup k naší MySQL databázi. Nastavení ODBC se provádí pomocí konfiguračního souboru odbc.ini. Tento konfigurační soubor se nachází v adresáři /etc/odbc.ini. Konfigurační soubor obsahuje údaje o umístění ovladačů databáze, adresu serveru, kde je umístěna databáze, a přihlašovací údaje databáze.

```
[ODBC Data Sources]
myodbc_ebrana   = MyODBC 3.52 Driver DSN

[myodbc_ebrana]
Driver       = /usr/lib/odbc/libmyodbc.so
Setup        = /usr/lib/odbc/libodbcmyS.so
Description  = MySQL ODBC 3.52 Driver DSN
Server       = localhost
Port         =
User         = uzivatel
Password     = heslo
Database     = databaze
Option       = 3
Socket       =
```

3.2.4 Test ODBC

Funkčnost rozhraní ODBC ověříme zadáním příkazu *odbctest* s parametry pro přihlášení k dané databázi.

```
odbctest DSN=databáze;UID=uživatel;PWD=heslo
```

Úspěšným výsledkem příkazu je spuštění příkazové řádky SQL, která umožňuje zadávat SQL dotazy a manipulovat s MySQL databází.

4 OpenLDAP

4.1 Instalace OpenLDAP

V této kapitole je popsána instalace produktu OpenLDAP na operačním systému Ubuntu[3] 9.10, který je distribucí systému Linux. Instalace OpenLDAP může probíhat dvěma způsoby.

První možností, jak nainstalovat server OpenLDAP, je provést instalaci pomocí balíků repositáře. V tomto případě můžeme celou instalaci zahájit příkazem v příkazovém řádku. Provedením příkazu *apt-get* se nainstaluje balíček obsahující daemona OpenLDAP serveru slapd a balíček ldap-utils, který obsahuje služby LDAP [6].

```
sudo apt-get install slapd ldap-utils
```

Po dokončení instalace můžeme zahájit konfiguraci serveru. Konfigurace OpenLDAP, nainstalovaného pomocí balíčků, se nachází v separované databázi *cn=config*, která má tvar DIT (viz. Jmenný model LDAP). V cn=config DIT lze dynamicky konfigurovat daemona slapd, bez nutnosti zastavit běžící službu LDAP. Celý proces konfigurace je dobře popsán na stránkách oficiální dokumentace k operačnímu systému Ubuntu[4].

U první možnosti instalace OpenLDAP se bohužel nedaří na serveru nakonfigurovat backend na databáze SQL. Je to zřejmě způsobeno faktem, že backend na databáze SQL je pouze experimentální, a tak ho distribuované balíčky OpenLDAP neobsahují. Z tohoto důvodu je potřeba zvolit druhou možnost instalace, u které sami kompilujeme zdrojové kódy.

Zdrojové kódy OpenLDAP nejlépe najdeme na domovských stránkách projektu[5]. Zde máme na výběr mezi několika vydanými verzemi. Nejvhodnější je ale zvolit poslední stabilní verzi. V našem případě je poslední stabilní verze OpenLDAP 2.4.21. Ke stažení aktuální verze můžeme využít nástroje pro stahování wget.

---

[3] http://www.ubuntu.cz

[4] https://help.ubuntu.com/8.10/serverguide/C/openldap-server.html

[5] http://www.openldap.org/software/download/

```
wget ftp://ftp.openldap.org/pub/OpenLDAP/openldap-
st..able/openldap-stable-20100219.tgz
```

Stažený archiv rozbalíme pomocí programu Gzip, kde rozbalení vyvoláme příkazem *gunzip*. Rozbalením souboru získáme balík, vytvořený programem Tar (Tape Archive), který rozdělíme na jednotlivé soubory pomocí příkazu *tar* a příznaků *–xf*, kde x (eXpand) znamená rozbalit a f (File) znamená soubor.

```
gunzip openldap-stable-20100219.tgz
tar -xvf openldap-stable-20100219.tar
```

Nyní můžeme vstoupit do adresáře, kde máme přístup k samotným souborům. Než začneme kompilovat, provedeme konfiguraci instalace. Tu provedeme spuštěním skriptu configure, který najdeme v adresáři. Všechny možnosti konfigurace získáme spuštěním skriptu s příznakem *--help*.

```
./cofigure --enable-sql --disable-hdb --disable-bdb
```

Konfiguraci spustíme s nastavením pro povolení databází SQL, což je důležité pro náš databázový backend, a zakázáním databází BDB a HDB, abychom se vyhnuli chybě, která hlásí absenci databáze tohoto typu. Při konfiguraci můžeme také narazit na chybu s chybějícími hlavičkovými souboru pro SQL. Chybu vyřešíme doinstalováním potřebných balíčků.

Po úspěšném dokončení konfigurace nám již zbývá vykonat poslední tři příkazy. Po jejich úspěšném provedení je instalace dokončena a můžeme začít s konfigurací serveru.

```
make depend
make
make install
```

Samotný příkaz pro instalaci je nutné, provést prostřednictvím privilegovaného uživatele. Proto je proveden společně z příkazem *sudo*, který slouží k provedení příkazu, přes uživatele root.

4.2 Konfigurace OpenLDAP

Konfigurace OpenLDAP je v tomto případě prováděna pomocí konfiguračních souborů. Pro konfiguraci serveru slouží soubor slapd.conf a pro konfiguraci klienta slouží soubor ldap.conf. Tyto soubory najdeme v závislosti na distribuci systému. V tomto případě se soubory nachází v adresáři /usr/local/etc/openldap/.

4.2.1 Soubor ldap.conf

Konfigurační soubor ldap.conf slouží k základní konfiguraci klientské části komunikace. Uživatelé si mohou sami vytvořit optimální konfigurační souboru ve svém domovském adresáři a toto základní nastavení nebude použito [3]. Soubor může obsahovat směrnici pro nastavení hodnoty adresy *URI*, která specifikuje adresu LDAP serveru, určení výchozího rozlišovacího jména směrnicí *BASE*, dále jsou zde umístěny směrnice pro použití zabezpečeného připojení TLS a směrnice pro nastavení mechanizmů SASL. Podrobnější popis jednotlivých směrnic najdeme na manuálových stránkách projektu OpenLDAP.

```
BASE        dc=ebrana,dc=cz
URI         ldap://127.0.0.1/
```

Hodnota adresy *URI* může obsahovat i nastavení portu. Standardně je to pro nezabezpečené připojení ldap://, port 389, a pro zabezpečené připojení ldaps://, port 636. V tomto případě je připojení nezabezpečené.

4.2.2 Soubor slapd.conf

Soubor slapd.conf obsahuje konfigurační informace procesu slapd. Proces slapd je stand-alone LDAP deamon, který vyčkává na připojení klienta [3]. Soubor se skládá z části globálního nastavení, které je platné pro celý server, a z části definic, která obsahuje specifikace pro databázový backend.

Globální nastavení

Hned v první části globálního nastavení najdeme důležitou část, bez které by se server neobešel. Tou je definování schémat, adresářového serveru. V tomto případě jsou zahrnuta tři schémata, která jsou standardně obsažena v inslatlaci OpenLDAP. První dvě schémata jsou definicí základních objektů LDAP adresáře. Schéma inetOrgPerson.schema nám už definuje konkrétní podobu záznamu pro zaměstnance organizace. Náhled objektové třídy inetOrgPerson je v kapitole Protokol LDAP – Informační model.

```
include  /usr/local/etc/openldap/schema/core.schema
include  /usr/local/etc/openldap/schema/cosine.schema
include
    /usr/local/etc/openldap/schema/inetorgperson.schema
```

Další směrnicí konfiguračního souboru je *LogLevel*. Tato direktiva slouží k nastavení úrovně ladění chyb a provoz statistik. Zde konkrétně je nastavena úroveň, kdy se zaznamenají statistiky připojení, operací a výsledků operací.

```
LogLevel 256
```

Do globálního nastavení je zahrnuta i část, kde jsou určena přístupová práva. Lze definovat standardní přístupová práva, která budou platit pro celý server, ale dají se konkretizovat na jednotlivé databázové vazby. Tento zápis nám udává přístupová práva pro všechny záznamy adresáře. Nastavení práv pro konkrétní atribut je popsáno v kapitole Bezpečnostní model – Autorizace.

```
access to *
    by dn="cn=root,dc=ebrana,dc=cz" write
    by anonymous auth
    by users read
    by self write
```

Mezi další možnosti globálního nastavení můžou patřit směrnice pro zabezpečení komunikace pomocí TLS, směrnice pro použití mechanizmu SASL, směrnice upřesňující vlastnosti navrácení výsledků operací atd. Podrobnější popis jednotlivých směrnic najdeme na manuálových stránkách projektu OpenLDAP.

Nastavení databáze

Tato část konfiguračního souboru slapd.conf obsahuje definice pro jednotlivé databáze, které server podporuje. Seznam podporovaných databází je uveden v Tabulce 1.

Tabulka 2 – Seznam podporovaných databází

Druhy	Popis
BDB	Berkeley DB
dnssrv	DNS server
HDB	Hierarchická varianta BDB
ldap	LDAP (Proxy) backend
meta	Meta Directory backend
monitor	Monitor backend
passwd	Poskytuje read-only přístupu k passwd
Perl	backend programovatelný v jazyce Perl
shell	Shell backend
sql	Backend programovatelný v jazyce SQL

zdroj: http://www.openldap.org/doc/admin24/slapdconfig.html

Některé směrnice jsou pro všechny typy databází stejné. Společnými směrnicemi pro všechny databáze jsou většinou povinné směrnice. První povinná směrnice *database* je určením typu databáze, v tomto případě tedy databáze SQL. Povinné je i určení adresářového kořenu, který určíme pomocí směrnice *suffix*. Pro danou databázi bychom měli taky určit administrátora. Směrnice *rootdn* a *rootpw* určují přihlašovací údaje administrátora dané databáze.

```
database      sql
suffix         "dc=ebrana,dc=cz"
rootdn         "cn=root,dc=ebrana,dc=cz"
rootpw         {MD5}XXXXX
```

U hodnoty hesla administrátora je ve složených závorkách údaj, který udává šifrovací metodu, kterou je ze zadaného hesla vytvořen hash. Heslo administrátora lze vygenerovat po instalaci pomocí příkazu:

```
slappasswd -h {MD5}
```

Dalšími společnými směrnicemi je určování vlastností databáze, například možnosti čtení z databáze, limity připojení, viditelnost databáze atp. Celý popis je detailněji na manuálových stránkách.

Zbylá konfigurace již obsahuje specifické směrnice pro konkrétní databáze. Jedná se hlavně o směrnice s umístěním databáze nebo směrnice s přístupovými údaji k databázi.

Například u databází typu BDB by byla použita směrnice *directory*, která určuje adresář pro umístění souborů dat.

```
directory     /usr/local/var/openldap-data
```

V našem případě se bude jednat od databázi typu SQL. Proto použijeme směrnice s přístupovými údaji k databázi. Přístup k databázi SQL zprostředkovává rozhraní ODBC. Pro vyplnění přístupu k databázi použijeme název databáze zadaný v rozhraním ODBC a přístupové údaje k databázi MySQL.

```
dbname          myodbc_ebrana
dbuser          uzivatel
dbpasswd        heslo
```

Další nastavení pro SQL databáze se týká úprav SQL dotazů. Zde můžeme pomocí směrnic měnit nebo upravovat dotazy, které slouží pro vyhledávání, vkládání, úpravu a mazání záznamů z databáze. Upravíme pouze možnosti vyhledávání směrnicí *subtree_cond*, protože databázi budeme používat pouze pro čtení (*readonly*). Dále pak zakážeme zaznamenávání údajů o práci s databází.

```
subtree_cond       "ldap_entries.dn LIKE CONCAT('%',?)"
has_ldapinfo_dn_ru no
lastmod            off
readonly           on
```

Dále lze nastavovat další zdroje dat, které budou tvořit další větve adresářového stromu. V tomto případě je již nastavení serveru dokončeno a můžeme ověřit správnost nastavení. Otestujeme konfigurační soubor provedením příkazu *slaptest*, pomocí kterého zjistíme, zda-li konfigurační soubor neobsahuje chyby.

5 Spuštění a použití serveru OpenLDAP

5.1 Spuštění serveru

Po úspěšné instalaci a konfiguraci OpenLDAP serveru můžeme server spustit. Spuštění provedeme spuštěním daemona Slapd. Soubor Slapd najdeme v tomto případě v adresáři */usr/local/libexec/*. Jeho spuštění provedeme příkazem s parametrem *start*.

```
sudo  /usr/local/libexec/slapd start
```

Bohužel po vykonání příkazu se nezobrazí žádná zpráva o úspěšném nebo neúspěšném spuštění serveru. Proto ověření zda příkaz proběhl bez problému a server je spuštěn zjistíme tak, že se mezi běžícím procesy objevil proces *slapd*. Jestliže vše proběhlo bez problému a server je spuštěn, můžeme zadávat dotazy a ověřit tak funkčnost adresáře.

V případě, kdy chceme činnost serveru zastavit použijeme příkazu pro „zabití“ procesu. Takovým případem může být provedení změny mapování v tabulkách databáze. Server se musí restartovat.

```
sudo  kill číslo_procesu
```

5.2 Použití adresáře

Nejjednodušší způsob, jak zadat dotaz adresáři, je využít předpřipravených nástrojů OpenLDAP, které jsem použil při popisu operací LDAP v kapitole Protokol LDAP Funkční model. Otestuji funkčnost vyhledání záznamů s jednoduchou autentizací, zadáním příkazu:

```
ldapsearch -b "uid=vonas,dc=ebrana,dc=cz" objectClass="*"
-D "cn=vonas,dc=ebrana,dc=cz" –w HESLO
```

Výsledkem příkazu je výpis záznamu uživatele *vonas*, ve formátu LDIF. Podoba výpisu je vidět na Obrázku 7.

```
# vonas, ebrana.cz
dn: uid=vonas,dc=ebrana,dc=cz
objectClass: inetOrgPerson
cn: Frantisek Vonasek
sn: Vonasek
uid: vonas
mail: frantisek.vonasek@ebrana.cz
givenName: Frantisek
employeeType: a
userPassword:: dm9uYXM=
employeeNumber: 3
```

Obrázek 7 – Výpis záznamu

Zdroj: Autor

Pro používání a správu adresáře, existuje i celá řada programů s grafickým uživatelským rozhraním. Nejpoužívanějšími jsou **phpLDAPadmin** a **Apache Directory Studio**.

Administrační aplikace phpLDAPadmin[6], je grafický nástroj pro správu adresáře, přes webové rozhraní. Aplikace využívá, pro přístup a správu, funkcí a příkazů jazyka PHP. Výhodou je možnost používání na mnoha platformách [12].

Software Apache Directory Studio[7], Java aplikace, vytvořená v prostředí Elipse. Tento produkt je primárně určen pro práci Apache Directory Server, cože jedna z implementací serveru LDAP [13]. Lze ho použít i pro server OpenLDAP. Tento software je ve firmě eBRÁNA aktivně využíván.

[6] http://phpldapadmin.sourceforge.net/wiki/index.php/Main_Page

[7] http://directory.apache.org/studio/

6 Využití LDAP webovou aplikací

Posledním krokem, je vytvoření jednoduché webové aplikace využívající adresář LDAP a test autentizace pomocí LDAP v aplikaci MantisBT . K vytvoření aplikace je použit jazyk PHP, který obsahuje funkce pro práci s adresářem LDAP.

6.1 Jazyk PHP

Jazyk PHP je skriptovací programovací jazyk, určený především k programování webových aplikací. Jazyk PHP je nezávislý na platformě. Syntaxe je inspirována jazyky C, Perl, Pascal a Java. Provádění skriptu PHP probíhá na straně serveru. K uživateli se dostane pouze výsledek provedeného skriptu. Jazyk PHP je velmi oblíbený pro svojí jednoduchost a velkou podporou knihoven, nebo modulů, pro různé účely (např. databáze SQL, LDAP) [14].

Abychom mohli využívat funkce, pro práci s adresářem LDAP, je nutné doplnit, náš interpret jazyka PHP, o knihovnu nebo modul, s těmito funkcemi. V tomto případě doplníme modul *php5-ldap*.

```
sudo apt-get install php5-ldap
```

6.1.1 Popis funkcí jazyka PHP

PHP obsahuje celé množství funkcí pro kompletní správu adresáře LDAP. V tomto případě, chceme adresář používat pro autentizaci uživatel, a pro získání záznamu, daného uživatele. Ve vytvořené aplikaci tedy využijeme pouze následující funkce:

- **ldap_connect()** – funkce pro navázání spojení se serverem;
    - Tato funkce může mít dva vstupní parametry. První je adresa serveru LDAP a druhý číslo komunikačního portu.
- **ldap_bind()** – funkce pro jednoduchou autentizaci uživatele u navázaného spojení;
    - Parametry je výsledek navázaného spojení a autentizační údaje uživatele (rozlišovací jméno a heslo).
- **ldap_search()** – slouží k vyhledání záznamu;

- Základními parametry jsou výsledek navázaného spojení, vychozí bod hledání a vyhledávací filtr.

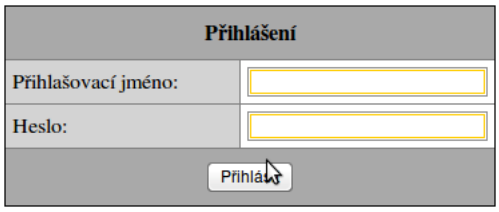
- **ldap_get_entries** – získání výsledku hledání záznamů;
  - Parametry jsou výsledek navázaného spojení a výsledek operace hledání. Výsledek je uložen do vícerozměrného pole.

- **ldap_close** – ukončení navázaného spojení;

Seznam a popis ostatních funkcí jazyka PHP pro protokol LDAP, lze najít na oficiálních stránkách jazyka PHP[8].

6.2 Autentizace uživatele webové aplikace

Vytvořená webová aplikace, obsahuje jednoduchý přihlašovací formulář (Obrázek 8), který slouží k ověření identity uživatele a získání informací o uživateli, které lze využít pro další práci s aplikací. Informace, které při ověřování získáme z adresáře, jsou uloženy do Session. Session je soubor proměnných pro danou relaci. Je to způsob jak v PHP uchovat proměnné, obsahující informace, například o právě přihlášeném uživateli.



Obrázek 8 – Náhled přihlašovacího formuláře

Zdroj: Autor

Celý proces přihlášení uživatele se provádí pomocí funkce *ldapLogin()*. Této funkci jsou předávány parametry, nesoucí údaje o přihlašovaném uživateli, informace potřebné k připojení LDAP serveru a pomocné údaje pro vyhledávání záznamu. V těle této funkce jsou použity všechny výše uvedené funkce PHP pro LDAP.

[8] http://www.php.net/

```
function ldapLogin($filter, $loginName, $loginPass, $baseDN, $server, $port) {
//tělo funkce
}
```

Výsledkem úspěšného přihlášení, se dostaneme na stránku přihlášeného uživatele, kde je výpis záznamu o uživateli. V této fázi máme všechny potřebné informace pro další běh aplikace.

Přihlášení	
Přihlášený uživatel:	
uid:	vonas
Celé jméno:	Frantisek Vonasek
Jméno:	Frantisek
Příjmení:	Vonasek
email:	frantisek.vonasek@ebrana.cz
Aktivita:	a
Úroveň privilegií:	3
Odhlásit...	

Obrázek 9 – Náhled přihlášeného uživatele

Zdroj: Autor

Z této fáze, můžeme směřovat běh aplikace dál podle potřeby. Pro různé aplikace, lze výpis atributů záznamu modifikovat. Společnost eBRÁNA chce takové formy přihlášení využívat například při administraci klientských stránek. Samotní uživatelé stránek budou přihlašováni pomocí databáze MySQL, ale administrátor z společnosti eBRÁNA bude mít přístup přidělen pomocí adresáře LDAP. Při takovém použití není potřeba vytvářet zvláštní účty pro administrátory a přístup mají pomocí vlastních přihlašovacích údajů.

6.3 Autentizace uživatele v systému MANTIS

MantisBT (Bug Tracking) je webová aplikace pro evidenci chyb v programech. Je to volná aplikace s licencí GNU General Public License[9] naprogramovaná v jazyce PHP. Využívá relačních databází SQL. Instalace na webový server může proběhnout na různých operačních systémech a většina webových prohlížečů je schopna Mantis správně zobrazovat. Aplikace Mantis nabízí možnost autentizace pomocí protokolu LDAP [16].

[9] všeobecná veřejná licence pro svobodný software

Aplikace Mantis je společností eBRÁNA využívána při vývoji svých aplikací. Jedná se o jednu z aplikací, která má využívat adresář LDAP, k přihlášení uživatel.

6.3.1 Nastavení aplikace Mantis

Pro nastavení webové aplikace Mantis, tak aby přihlášení uživatel proběhlo pomocí adresáře LDAP, je potřeba změnit několik proměnných konfiguračního souboru *config_default_inc.php*. V souboru nastavíme hodnoty oddíl proměnných pro nastavení připojení k LDAP.

```
$g_ldap_server          =    'ldap://localhost/';
$g_ldap_port            =    389;
$g_ldap_root_dn         =    'dc=ebrana,dc=cz';
$g_ldap_organization    =    '';
$g_ldap_uid_field       =    'uid';
$g_ldap_realname_field  =    'cn';
$g_ldap_bind_dn         =    'cn=root,dc=ebrana,dc=cz';
$g_ldap_bind_passwd     =    'heslo';
$g_use_ldap_email       =    ON;
$g_use_ldap_realname    =    ON;
$g_ldap_protocol_version =   0;
$g_ldap_follow_referrals =   OFF;
$g_ldap_simulation_file_path = '';
```

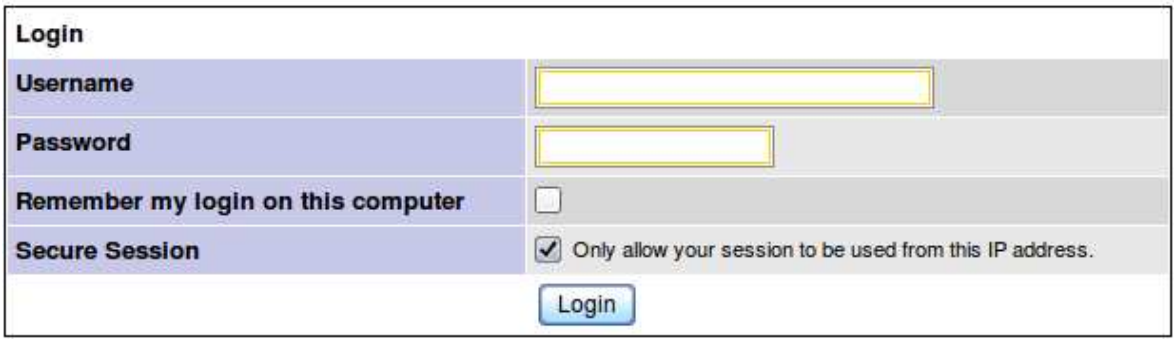
Nakonec změníme hodnotu proměnné pro určení způsobu autentizace. Máme několik možností CRYPT, PLAIN, MD5, LDAP nebo BASIC_AUTH.

```
$g_login_method    =    LDAP;
```

6.3.2 Přihlášení do aplikace Mantis

Přihlašovací údaje, které uživatel zadá, jsou porovnány s výsledkem vyhledávání záznamů z adresáře LDAP. Na základě shody údajů s výsledkem vyhledávání, je uživatel přihlášen nebo ne. Vzhledem k tomu, že aplikace plně využívá relační databázi, v tomto případě je to databáze MySQL, je potřeba udržovat údaje o uživateli i v této databázi. U straších verzí Mantisu je to řešeno registrací uživatele v aplikaci Mantis a poté vytvoření záznamu uživatele v adresáři LDAP. V poslední stabilní verzi Mantis 1.2.1, je řešení již vyřešeno daleko pohodlněji. V případě kdy vytvoříme záznam v adresáři a přihlásíme se k aplikaci Mantis, je záznam o uživateli v tabulkách databáze vytvořen automaticky z dat

adresáře. V případě společnosti eBRÁNA již tabulka uživatelů aplikace Mantis obsahuje uživatelská data. Pro tento případ Mantis provede porovnání uživatelských jmen, nebo jiného rozlišovacího jména uživatele.



Obrázek 10 – Náhled přihlašovacího formuláře aplikace Mantis

Zdroj: Autor

Závěr

Při zpracování této bakalářské práce byla prozkoumána možnost spojení adresářového serveru LDAP s relační databází SQL. Na základě toho bylo nutné upravit databázi MySQL, zprovoznit a nakonfigurovat rozhraní ODBC pro databázi MySQL, nainstalovat a nakonfigurovat software OpenLDAP, vytvořit testovací webovou aplikaci a otestovat použití u webových aplikací.

V práci bylo dosaženo stanovených cílů:

- Zprovoznění adresářového serveru, který využívá relační databázi SQL;
- Vytvoření testovací webové aplikace;
- Otestování připojení k adresářovému severu přes webové rozhraní;

Všechny dosažené cíle byly úspěšně vyzkoušeny v prostředí společnosti eBRÁNA ve spolupráci s vedoucím práce a IT pracovníkem společnosti. Výsledkem je usnadnění práce s uživatelskými účty zaměstnanců společnosti. To šetří čas i finanční prostředky společnosti eBRÁNA.

V souvislosti s plněním zadaného úkolu jsem se naučil konfigurovat a používat rozhraní ODBC. Naučil jsem nainstalovat a konfigurovat adresářovou službu OpenLDAP. Rozšířil jsem znalosti o programovacím jazyku PHP, využíváním funkcí pro přístup k adresářovému serveru LDAP a seznámil jsem se s pracovním prostředím společnosti eBRÁNA. Získané zkušenosti lze využít v budoucím zaměstnání.

V budoucnosti bude adresářový server využit jako součást informačního systému. Pro takové nasazení je určitě vhodné rozšířit adresářovou službu o zabezpečení komunikace s adresářovým serverem pomocí TSL/SSL. Dále lze rozšířit působnost adresářového serveru pro více aplikací a služeb, například pro síťový protokol Samba, a rozšířit další obsah adresáře.

Literatura a zdroje

[1] BANÁK, Karel. *Použití adresářových služeb v informačních systémech* [online]. Praha, 2004. 77 s. Diplomová práce. ČVUT, FS, Katedra systémového inženýrství. Dostupné z WWW: <http://ldap.benak.net/diplom.pdf>.

[2] LDAP In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 12.3.2006, 26.2.2010 [cit. 2010-04-11]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/LDAP>.

[3] The OpenLDAP Foundation. *OpenLDAP : community developed LDAP software* [online]. c2010 [cit. 2010-05-06]. Dostupné z WWW: <http://www.openldap.org/>.

[4] D. DENT, Kyle. *Postfix : kompletní průvodce*. Praha : Grada, 2005. 252 s. ISBN 80-247-1029-3.

[5] Active Directory In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 31. 7. 2005, 21. 4. 2010 [cit. 2010-04-28]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Active_Directory>.

[6] *Official Ubuntu Documentation : OpenLDAP Server* [online]. 2008 [cit. 2010-04-30]. Ubuntu Documentation. Dostupné z WWW: <https://help.ubuntu.com/8.10/serverguide/C/openldap-server.html>.

[7] *Fefes Homepage* [online]. 2002-04-03 [cit. 2010-05-03]. Tinyldap - a small LDAP implementation. Dostupné z WWW: <http://www.fefe.de/tinyldap/>.

[8] *MySQL In Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 21.11.2004, 7.4.2010 [cit. 2010-05-04]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/MySQL>.

[9] LDAP Data Interchange Format In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 27.5.2007, 21.2.2010 [cit. 2010-05-05]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/LDAP_Data_Interchange_Format>.

[10] Open Database Connectivity In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 7.12.2006, 17.11.2009 [cit. 2010-05-05]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Open_Database_Connectivity>.

[11] *Flat Mountain : Saving computers, one person at a time...* [online]. August 5th, 2004 [cit. 2010-05-05]. Setting up LDAP with back-sql. Dostupné z WWW: <http://www.flatmtn.com/article/setting-ldap-back-sql#LdapGeneral-10>.

[12] *PhpLDAPadmin* [online]. 6 October 2008, 4 February 2010 [cit. 2010-05-06]. Main Page. Dostupné z WWW: <http://phpldapadmin.sourceforge.net/wiki/index.php/Main_Page>.

[13] BENÁK, Karel. *Benak-Net : Software pro správu LDAP serveru* [online]. 2007-05-31 [cit. 2010-05-06]. Apache Directory Studio. Dostupné z WWW: <http://www.benak.net/ldap/ldapstudio>.

[14] PHP In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2. 6. 2004, 4. 5. 2010 [cit. 2010-05-06]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/PHP>.

[15] *Php.net* [online]. c2001-2010 [cit. 2010-05-06]. LDAP - Manual. Dostupné z WWW: <http://php.net/manual/en/book.ldap.php>.

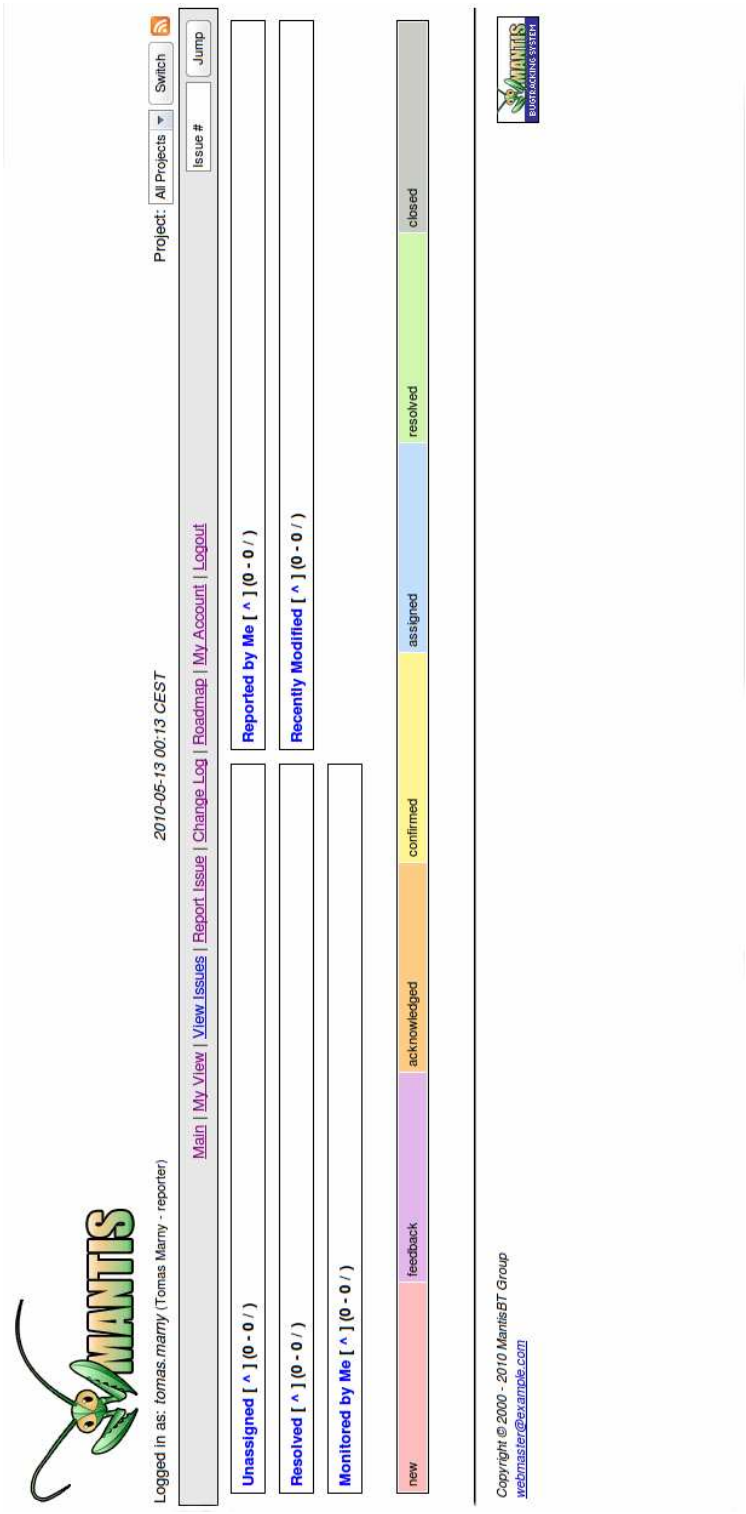
[16] *Mantis : Bug Tracker* [online]. c2000-2010 [cit. 2010-05-06]. Dostupné z WWW: <http://www.mantisbt.org/>.

Příloha A – Obsah přiloženého CD

Přiložené CD obsahuje:

- Dokument *bakalarska_prace.pdf*, obsahující text bakalářské práce v elektronické podobě;
- Archiv *LDAP_login.zip*, obsahující testovací webovou aplikaci;
- Model databáze uživatelů a tabulek LDAP(soubor *MySQL_LDAP_Backend.txp*);
- Archiv *openldap-stable-20100219.tgz* s aplikací OpenLDAP verze 2.4.21 a archiv *mantisbt-1.2.1.zip* obsahující webovou aplikaci MantisBT;

Příloha B – Rozhraní aplikace Mantis



Obrázek 11 – Úvodní strana aplikace Mantis

Zdroj: Autor

Obrázek 12 – Údaje o uživateli aplikace Mantis

Zdroj: Autor