UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Návrh a konfigurace poštovního a souborového serveru na platformě GNU/Linux

Vít Šretr

Bakalářská práce
2012

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2011/2012**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Vít Šretr**
Osobní číslo:	**I09280**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Návrh a konfigurace poštovního a souborového serveru na platformě GNU/Linux**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování:

Cílem práce je navrhnout a nakonfigurovat linuxový server, jehož hlavní služby budou zabezpečeny prostřednictvím Postfix a Samba serveru. Autor práce představí problematiku MTA, principy komunikace a možnosti nasazení Postfix jako MTA klienta. Autor dále představí problematiku síťového protokolu SMB a možnosti jeho využití na linuxových serverech s využitím technologie Samba. Autor provede reálnou konfiguraci serveru přizpůsobenou pro využití v rámci laboratoře při výuce předmětu počítačové sítě 4.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

DENT, K. D. Postfix : kompletní průvodce. 1. Praha : Grada, 2005. 252 s. ISBN 80-247-1029-3.
TYS, J.; ECKSTEIN, R.; COLLIER-BROWN, D. Using Samba. 2. [s.l.] : O'Reilly Media, 2003. 560 s. ISBN 978-0596002565.

Vedoucí bakalářské práce: **Mgr. Josef Horálek**
Katedra softwarových technologií

Datum zadání bakalářské práce: **16. prosince 2011**
Termín odevzdání bakalářské práce: **11. května 2012**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 30. března 2012

Prohlášení autora

Prohlašuji, že jsem tuto práci 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 27. 04. 2012 Vít Šretr

Poděkování

Chtěl bych tímto poděkovat vedoucímu své bakalářské práce Mgr. Josefu Horálkovi za věnovaný čas a odborné rady, které mi výrazně pomohly při tvorbě práce. Dále bych chtěl poděkovat mé přítelkyni Mgr. Janě Khýnové a celé rodině za podporu během doby mého studia.

Anotace

Tato práce se zabývá základními požadavky na serverový operační systém, obecnou architekturou linuxového jádra a konfigurací poštovního a souborového serveru na platformě GNU/Linux se zaměřením na použití při výuce předmětu počítačové sítě 4.

Klíčová slova

server, linux, gnu, postfix, samba, kerberos, ldap

Title

GNU/Linux mail and file server configuration

Annotation

This bachelor's thesis is dedicated to describe basic requirements for the server operating system, the architecture of the Linux kernel and configuration of GNU/Linux mail and file server. This configuration will be used for teaching.

Keywords

server, linux, gnu, postfix, samba, kerberos, ldap

Obsah

Seznam zkratek

APT	Advanced Packaging Tool
ASCII	American Standard Code for Information Interchange
AT	Advanced Technology
BDC	Backup Domain Controller
CIFS	Common Internet File System
CPU	Central Processing Unit
CUPS	Common Unix Printing System
DC	Domain Component
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DN	Distinguished Name
DNS	Domain Name Server
DOS	Disk Operating System
DVD	Digital Versatile Disc
FTP	File Transfer Protocol
GCC	GNU Compiler Collection
GIMP	GNU Image Manipulation Program
GNU	GNU's Not Unix
GPL	General Public License
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IMAP	Internet Message Access Protocol
IP	Internet Protocol
ISP	Internet Service Provider
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
LMTP	Local Mail Transfer Protocol
MDA	Mail Delivery Agent
MID	Multiple ID
MIME	Multipurpose Internet Mail Extensions
MS	Microsoft
MTA	Mail Transfer Agent
MUA	Mail User Agent
MX	Mail Exchanger
NetBIOS	Network Basic Input Output System
NFS	Network File System
NID	Network ID
NIS	Network Information Service
NT	New Technology

NTP	Network Time Protocol
OS	Operating System
OS/2	Operating System/2
OU	Organization Unit
PDC	Primary Domain Controller
PDF	Portable Document Format
PID	Process ID
POP	Post Office Protocol
POSIX	Portable Operating System Interface
PPD	PostScript Printer Description
RAM	Random Access Memory
RFC	Request for Comments
RMS	Richard Matthew Stallman
SAM	Security Accounts Manager
SASL	Simple Authentication and Security Layer
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SSO	Single sign-on
SWAT	Samba Web Administration Tool
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UID	User ID
UNC	Uniform Naming Convention
WINS	Windows Internet Naming System

Seznam obrázků

Úvod

V oblasti serverových síťových služeb je elektronická pošta jednou z nejvíce používaných služeb na světě. Správné nastavení poštovního serveru je velice obtížná úloha, jelikož zejména server Postfix je vysoce flexibilní a může být použit pro nejrůznější účely, které se elektronické pošty týkají.

Souborový server je velmi užitečný zejména v prostředí, kde se uživatelé pohybují mezi počítači a potřebují mít svá data vždy k dispozici. Tento problém řeší server Samba, který nabízí možnost sdílení adresářů i tiskáren. Vhodné nastavení Samba serveru je komplexní problematika, zejména pokud je požadována spolupráce s centrální autentizační databází. U obou serverů existuje nespočet různých direktiv, které ovlivňují chování serveru. Tato práce si bere za cíl seznámit studenty v rámci výuky předmětu Počítačové sítě se způsobem fungování a základním nastavením těcho serverů.

První část práce je zaměřena na obecné seznámení s požadavky, které jsou kladeny na serverové operační systémy. Jednoduše lze říci, že nejdůležitejší vlastnosti serverového operačního systému jsou bezpečnost a stabilita.

Část druhá je věnována problematice unixových operačních systémů, jmenovitě systému GNU/Linux. Důraz je kladen hlavně na základní architekturu linuxového jádra a způsob, jakým jádro zavádí moduly.

Jako operační systém pro tuto práci byl zvolen Debian GNU/Linux squeeze 6.0. Tato linuxová distribuce je již skoro dvacet let používána pro serverové nasazení a má za sebou velmi širokou komunitní podporu vývojářů. Díky svému specifickému vývojovému cyklu poskytuje vysoce stabilní verze systému vhodné pro servery.

Třetí kapitola se věnuje poštovnímu serveru Postfix ve verzi 2.7.1. Úvodní část obsahuje seznámení s formátem a komponentami elektronické pošty, s protokoly, které elektronická pošta používá a způsobem putování zprávy po síti Internet od odesilatele k příjemci. Dále jsou popsány základní vlastnosti serveru Postfix a průchod elektronické zprávy serverem. Závěr třetí kapitoly popisuje základní nastavení Postfixu jako MTA a nastavení programu Courier, který slouží jako server MDA pro vyzvedávání doručených zpráv uživateli.

V poslední části je práce věnována souborovému serveru Samba verze 3.5.6. Jsou popsány základní pojmy, se kterými se uživatel setká při konfiguraci, a také protokoly LDAP a Kerberos, které ve vzájemném spojení tvoří velmi silný nástroj pro centrální správu uživatelů a jednoduchý přístup k síťovým službám. Druhá polovina čtvrté kapitoly popisuje základní nastavení sdílení adresářů a tiskáren v rámci sítě LAN. Pozornost je též zaměřena na konfiguraci centrálního autorizačního serveru uživatelů pomocí již zmíněných služeb LDAP a Kerberos.

1 Úvod do serverových operačních systémů

1.1 Co je to OS?

Operační systém umožňuje ovládat základní řízení prostředků počítače a umožňuje jeho využívání prostřednictvím uživatelských programů. Mezi tyto prostředky patří procesor, operační paměť, vstupně-výstupní zařízení a soubory. Základní struktura operačního systému se sestává z jádra a pomocného programového vybavení. Holý počítač bez operačního systému je pro běžného uživatele nepoužitelný, a proto OS tvoří vrstvu mezi uživatelskými programy a hardwarem počítače [1].

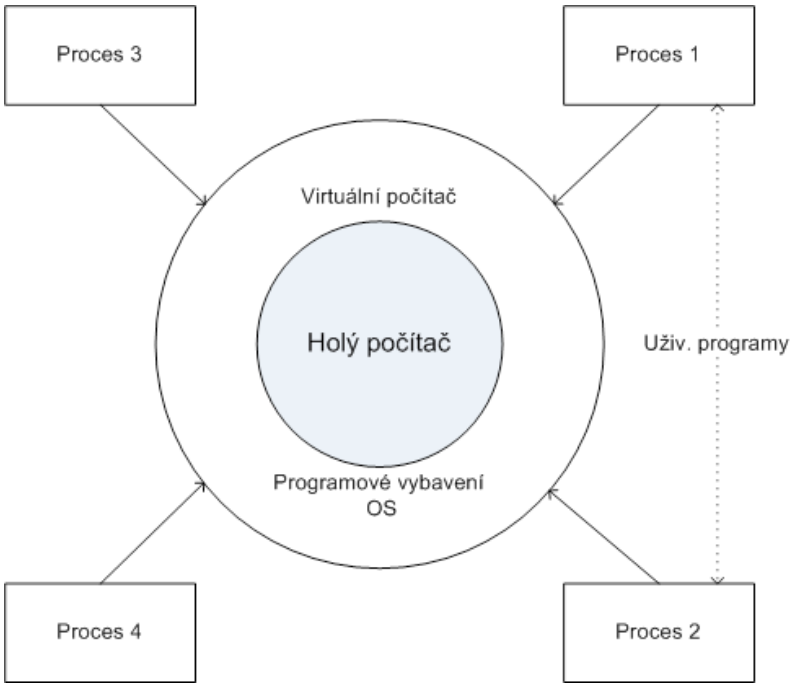
1.2 Funkce OS

1.2.1 Správce prostředků

Správce prostředků je funkce operačního systému, která se stará o přidělování a odebírání sdílených systémových prostředků jednotlivým procesům [2]. Správce prostředků používá tzv. multiplexing pro sdílení prostředků v čase (CPU) a v prostoru (RAM).

1.2.2 Rozšíření stroje – virtualizace

Operační systém ukrývá před programátorem detaily ovládání jednotlivých zařízení a definuje standardní rozhraní pro volání systémových služeb. Programátor tedy není zatěžován prací se vstupně-výstupními zařízeními a může se naplno věnovat vlastní úloze. Tím operační systém umožňuje ovládání počítače na přijatelné úrovni [2].



Obrázek 1 – Virtuální počítač [1]

1.3 Požadavky na serverový OS

Nejzákladnějším požadavkem na serverový operační systém je jeho stabilita. U serveru je naprosto nezbytné minimalizovat dobu, po kterou není schopen provozu kvůli chybě způsobené operačním systémem. Vhodně zvoleným operačním systémem lze tuto dobu významně zredukovat. Neméně důležitou vlastností serverového OS je jeho odolnost vůči pokusům útočníků server poškodit či dokonce vyřadit z provozu. Pro server, který je exponován vysokému počtu požadavků ze strany klientů, je nutný vysoký výpočetní výkon. Ten je zajištěn správně dimenzovaným hardwarovým vybavením, které musí serverový OS podporovat ve svém jádře [3].

Na poli serverových OS se utkávají dva hlavní rivalové. Produkty řady Windows Server od firmy Microsoft a systémy unixového typu. Unixové systémy, zejména GNU/Linux, jsou často preferovány jako vhodné serverové řešení kvůli své modularitě, spolehlivosti a také ceně, jelikož se jedná o volně šiřitelný software [4].

1.4 Služby poskytované serverovým OS

Jen stěží si lze představit fungování jakékoliv počítačové sítě bez serveru, který poskytuje služby klientům na základě modelu klient-server. Mezi nabízené služby může například patřit prezentování webových stránek pomocí protokolu HTTP, sdílení souborů a tiskáren protokolem SMB, zasílání zpráv elektronické pošty skrze protokoly POP3, SMTP a IMAP4 či komplexní ověřování přístupových práv uživatelů k síťovým službám doménovým řadičem Samba. Přidělování potřebných konfiguračních nastavení stanic v síti (DHCP) a poskytovaní překladu doménových jmen na IP adresy pomocí DNS serveru mohou být též realizovány jako serverové služby, avšak jsou běžně poskytovány moderními směrovači [3].

2 Popis architektury GNU/Linux

2.1 Historie

Projekt GNU byl založen roku 1984 Richardem Matthew Stallmanem. Jeho cílem bylo vytvořit operační systém unixového typu, jenž by disponoval pouze svobodným softwarem. Jako jádro svého operačního systému si RMS (tak je Stallman znám v komunitě okolo hnutí GNU) zvolil Hurd, jehož vývoj byl započat roku 1990. V té době již GNU disponovalo všemi důležitými aplikacemi, systémovými knihovnami, překladačem jazyka C (GCC), textovým editorem a dalším softwarem. Ale jelikož byl Hurd navržen jako mikrojádro, které se velmi složitě implementuje, je již přes 20 let v aktivním vývoji a dosud nebyla zveřejněna žádná oficiální verze.

Původem slova GNU je rekurzivní akronym GNU's Not Unix (GNU není Unix). Též je to anglický výraz pro pakoně hřivnatého. Ten je vyobrazen v logu projektu a byl pojmenován jako Heckert.

Zcela nezávisle začal na druhém konci světa vývoj vlastního unixového jádra dle standardu POSIX finský student univerzity v Helsinkách Linus Torvalds. Inspirací mu byl operační systém systém Minix, s nímž se seznámil na půdě univerzity. K Minixu však nebylo možno získat zdrojové kódy a ostatní unixové systémy byly pro studenta cenově nedostupné. Proto se Torvalds rozhodl vytvořit si vlastní operační systém v jazyce C podobný Minixu, který by byl provozovatelný na běžném osobním počítači.

Do usenetu (síťový komunikační systém) tehdy poslal dnes již legendární zprávu:

„Hello everybody out there using minix -I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones.“ [5]

Okolo jádra se okamžitě vytvořila masivní komunita, která začala na vývoji spolupracovat. Rychlost vývoje předčila veškerá Torvaldsova očekávání, a jelikož jádro Hurd bylo stále naprosto nepoužitelné, došlo k logickému kroku a operační systému GNU se v roce 1992 začal distribuovat s jádrem Linux. Tím vznikl operační systém se správným označením GNU/Linux.

Původní název Linusova jádra měl být Freax, spojení slov free (svodobný), freak (šílený) a unix. Avšak Torvaldsovův přítel, Ari Lemmke, při zveřejnění jádra ve verzi 0.01 na FTP serveru změnil na jeho počest název na Linux (spojení slov Linus a unix). Maskotem projektu Linux je tučňák Tux, jenž byl nakreslen v grafickém programu GIMP až v roce 1996. Původně měl být Tuxův vzhled drsný a agresivní, ale nakonec zvítězila podoba spokojeného a kamarádského Tuxe.

Linus Torvalds má stále hlavní slovo ve vývoji Linuxu, i když současné jádro obsahuje méně než 2% jeho kódu. Richard Matthew Stallman se věnuje propagaci svobodného softwaru, práci pro GNU Fundation a vývoji textového editoru GNU Emacs [5].

2.2 Licence GNU GPL

GNU General Public License (všeobecná veřejná licence) je softwarová licence pro svobodný software, která byla vytvořena Richardem Stallmanem pro projekt GNU. Řeší především přístupnost a distribuci zdrojových kódů softwaru. V případě, že je software distribuován pod touto licencí, je zaručeno, že všechny vydané vylepšené verze budou též svobodným softwarem. Jedná se tedy o copyleftovou licenci, která vyžaduje, aby verze programů, které jsou licencovány pod GPL a jsou dále modifikovány, zůstaly pod touto licencí [6].

Všechny části systému GNU i jádra Linux jsou šířeny pod licencí GNU GPL. Důležité je si uvědomit, že ne veškerý svobodný software musí být distribuován pod licencí GNU GPL. Richard Stallman při zrodu projektu GNU definoval 4 základní svobody pro všechny uživatele svobodného softwaru:

- Svoboda spustit program za libovolným účelem.
- Svoboda studovat, jak program pracuje a přizpůsobit ho svým požadavkům.
- Svoboda redistribuovat kopie, abyste pomohli kolegovi.
- Svoboda vylepšovat program a zveřejňovat vylepšení, aby z nich mohla mít prospěch celá komunita [6].

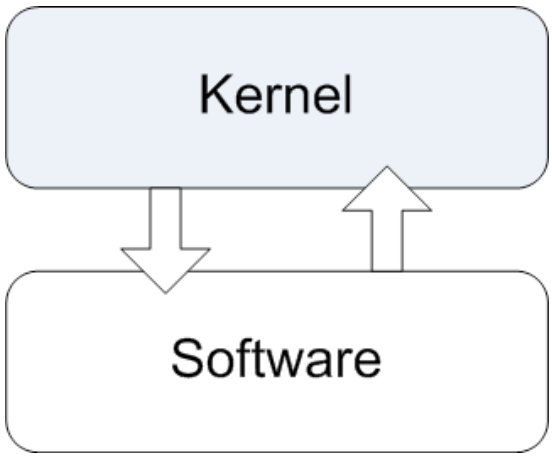
2.3 Linuxové distribuce

Operační systém GNU/Linux je šířen formou tzv. distribucí. Neexistuje tedy jediný systém GNU/Linux, ale každý má možnost vzít si jádro Linuxu, software a knihovny projektu GNU a další základní aplikační software (GUI, kancelářský balík, atd.) a postavit z nich vlastní distribuci [7]. Distribucí existuje skutečně mnoho, přičemž každá míří do svého sektoru uplatnění, ať už se jedná o serverové či desktopové nasazení, nebo o použití ve vestavěných systémech. Distribuce by měly být kompatibilní se standardem Linux Standard Base [4].

2.4 Monolitické jádro

Linux je vyvíjen jako tzv. modulární monolitické jádro. Veškerý kód takového jádra je rozdělen na základní (samotné jádro) a moduly (ovladače). Linuxové jádro běží celé ve stejném adresním prostoru a podporuje načítání externích modulů. Tento návrh jádra poskytuje zrychlení běhu, zmenšení velikosti jádra samotného i jeho paměťových nároků a možnost připojovat a odpojovat moduly za běhu systému. Další výhodou jádra Linux je, narozdíl od běžných monolitických jader, možnost uplatňovat, za určitých podmínek, preemptivní multitasking i na moduly. Mezi nevýhody modulární architektury patří provázanost jednotlivých částí jádra. Tím pádem může chyba v jedné části ovlivnit část jinou nebo dokonce celé jádro [8].

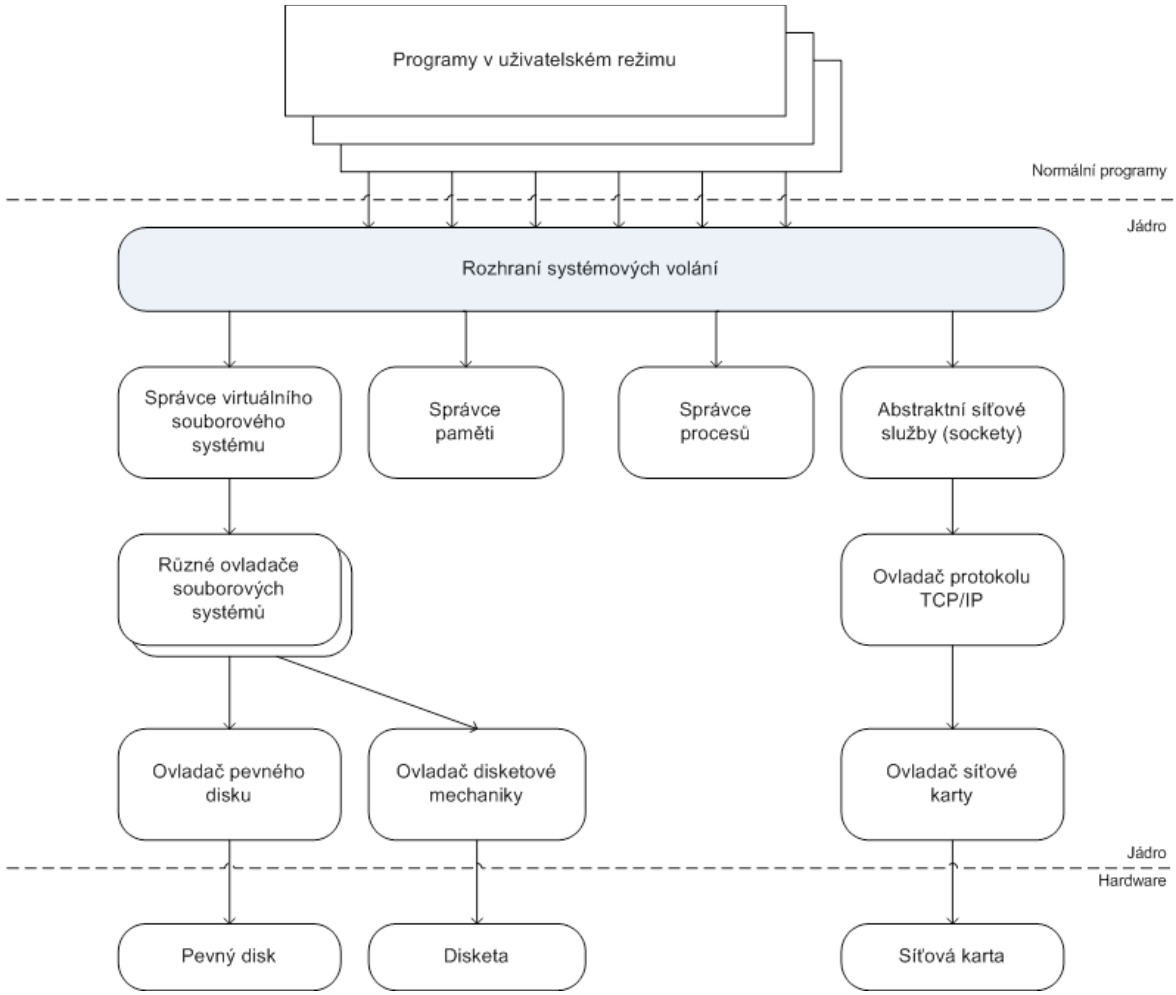
Právě monolitický návrh Linuxu byl předmětem střetu názorů Linuse Torvaldse a Andyho Tanenbauma, který kritizoval fakt, že Linux není mikrojádro. Mikrojádro implementuje vetšinu činností monolitického jádra v procesech, které běží v uživatelském paměťovém prostoru [8].



Obrázek 2 – Monolitické jádro [8]

Jádro Linuxu se skládá z několika důležitých podsystémů. Mezi nejdůležitější subsystémy patří správce procesů a správce paměti. Správce procesů vytváří procesy a pomocí plánovače přepíná mezi aktivními procesy [4]. Tím určuje, který proces může využívat procesor, a tím zabezpečuje multitasking, který poskytuje uživateli možnost pseudoparalelního běhu libovolného počtu procesů.

Správce paměti na druhou stranu přiděluje a odebírá procesům paměťové a odkládací oblasti. Dále zajišťuje virtuální adresování, obvykle implementované stránkováním nebo segmentací, díky kterému může mít každý proces vlastní virtuální adresový prostor. To umožňuje izolovat jednotlivé procesy navzájem a ochránit jádro před procesem, který by požadoval alokaci paměti, na níž se nachází paměťový prostor jádra [8].



Obrázek 3 – Důležité části jádra Linuxu [4]

Linux na nejnižší úrovni obsahuje ovladače pro všechna základní zařízení, která operační systém podporuje. Jelikož existuje velmi široká škála různých druhů hardwaru, je počet těchto ovladačů zařízení vysoký. Obecné třídy ovladačů využívají podobnosti mezi zařízeními a každý člen takové třídy disponuje rozhraním k ostatním částem jádra. Díky tomu vypadají například všechny ovladače disků pro zbytek jádra podobně a liší se pouze implementací operací skrytých pod rozhraním [4].

2.5 Moduly jádra

Jaderné moduly umožňují přidat kód do jádra i za jeho běhu. U nemodulárního jádra je tedy nutné přidat ke zdrojovému kódu jádra další zdrojové soubory a jádro znovu přeložit. Zaváděné moduly byly do jádra implementovány až v roce 1995. Jaderné moduly slouží především pro zavádění ovladačů zařízení, ovladačů souborového systému a systémových volání. Není tudíž nutno po přidání nového hardwaru restartovat celý systém. Stačí pouze nahrát potřebné moduly do jaderného adresového prostoru. Tím pádem se jádro stává flexibilním vůči změnám ovladačů. Nahrání modulů vyžaduje jistou režii v podobě časového zpoždění, které je ovšem zanedbatelné [4].

Základní nástroje pro vkládání a odstraňování modulů jsou programy *insmod* a *rmmod* [4].

Po spuštění příkazu *insmod*, jehož parametrem je název modulu, se na pozadí operačního systému otevře soubor s modulem, ten se překopíruje do paměti, zavolá se systémové volání *init_module,* paměť se uvolní a soubor se uzavře [9].

Příkaz *rmmod*, který zavolá systémové volání *delete_module*, slouží pro odstranění zavedeného modulu. Následuje kontrola oprávnění uživatele odstraňovat moduly, vyhledání modulu v seznamu, kontrola případných závislostí mezi ostatními moduly a možnosti modul odebrat, čekání na uvolnění modulu, zavolání úklidové funkce a odebrání modulu ze seznamu včetně uvolnění alokovaných prostředků [9].

2.6 Démoni – služby na pozadí

Démoni jsou trvale, v pozadí, běžící procesy. Charakteristickým znakem je písmeno *d* na konci názvu procesu. Typicky jsou spouštěni při startu operačního systému a čekají na události, na které reagují a poskytují příslušné služby [4]. Zpravidla nedisponují ani uživatelským rozhraním, jelikož jejich činnost využívají jiné procesy, vzdáleně přihlášení uživatelé nebo jádro [2].

3 Poštovní server – Postfix

3.1 Elektronická pošta

Historie internetové elektronické pošty (emailu) sahá až do počátků 70. let 20. století, kdy bylo přes síť Arpanet[1] možno odesílat první elektronické zprávy [10]. Mezi přednosti elektronické pošty patří samotná elektronická forma, která eliminuje náklady spojené s poštou klasickou, dále pak možnost zautomatizování kroků nutných ke zpracování dané zprávy a v neposlední řadě zvyšuje uživatelský komfort jak odesilatele, tak adresáta. Postupem času, a s rostoucí oblibou emailu, došlo k jeho vylepšení v několika směrech. Byla zavedena podpora různých znakových sad, sjednocen způsob přibalování příloh a též se rozšířil repertoár formátů, jenž může obsahovat samotné tělo zprávy [12]. Služby internetové pošty rozšířil mezi veřejnost Hotmail[2] v roce 1996. Od té doby je tento způsob internetové komunikace nejpoužívanější.

3.1.1 Komponenty elektronické pošty

Procesu přenosu emailu se účastní mnoho různých softwarových součástí. Každá se přitom stará o jiný krok procesu dodání zprávy do cíle [10].

MUA

Software, který zná většina uživatelů elektronické pošty, je označován jako MUA (Mail User Agent). MUA poskytuje základní uživatelský komfort pro čtení a psaní zpráv, avšak o samotné dodání zprávy do cíle se téměř nezajímá. Nejznámějšími agenty MUA jsou mutt, Outlook Express, KMail a Mozilla Thunderbird [10].

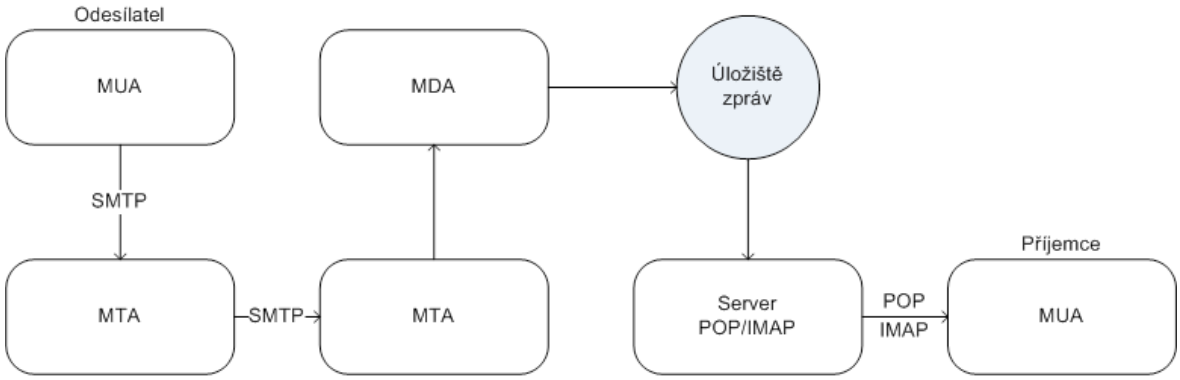
MTA

Agent MTA (Mail Transfer Agent) přebírá zprávy od MUA a má na starosti veškerou práci týkající se odesílání a přijímání uživatelských zpráv. Příkladem agentů MTA jsou Qmail, Sendmail a Postfix [3], kterému bude věnována tato kapitola.

MDA

MDA (Mail Delivery Agent) je program, jenž se specializuje na místní zpracování a uložení pošty. Komunikace těchto agentů již typicky neprobíhá po síti, ale jen přijatou zprávu uloží. Může též provádět antivirovou či antispamovou kontrolu. Typickými zástupci MDA jsou Procmail nebo Maildrop [3].

[1] Advanced Research Projects Agency Network, předchůdce dnešního Internetu.
[2] Zprostředkovatel internetové pošty pod záštitou firmy Microsoft.



Obrázek 4 – Jednoduchý průchod zprávy Internetem [10]

Obrázek 4 znázorňuje jednoduchý průchod zprávy Internetem. Uživatel pomocí MUA napíše zprávu, ten ji předá poštovnímu serveru, který disponuje MTA. Agenti MTA přebírají veškerou zodpovědnost za přesun zprávy z jednoho systému na druhý. V případě, že MTA obdrží požadavek na příjem zprávy, stanoví, zda má danou zprávu přijmout či nikoli. MTA obvykle přijímá zprávy pro své lokální uživatele nebo pro jiné systémy, kterým umí zprávu předat. Po přijetí se musí rozhodnout, co s danou zprávou provede. Pokud zpráva nepatří uživateli na jeho lokálním systému, předá ji dál jinému agentovi MTA. Zprávy určené pro jiné sítě mohou projít mnoha systémy. V případě, že MTA nedokáže zprávu doručit ani ji předat dále, odešle ji zpět původnímu odesílateli nebo na tuto situaci upozorní správce systému, kterým může být poskytovatel připojení k Internetu (ISP) v případě jednotlivců či podnikové oddělení informačních systémů v případě zaměstnanců.

Na konci cesty dorazí zpráva na MTA, které je jejím konečným cílem. Pokud zpráva náleží uživateli tohoto systému, MTA ji předá agentovi MDA, a ten zajistí její konečné doručení. MDA uloží zprávu do uložiště zpráv, které může být i speciální databází.

Zpráva je uložena v uložišti do doby, kdy je příjemce připraven k jejímu vyzvednutí. Ten posléze použije k převzetí zprávy agenta MUA, který zkontaktuje server poskytující přístup k uložišti zpráv. Pokud je uživatel u serveru úspěšně ověřen, může být zpráva předána agentovi MUA a přečtena adresátem [10].

3.1.2 Protokoly elektronické pošty

Pro fungování doručování zpráv elektronické pošty jsou používány tři hlavní protokoly. Protokol SMTP pro odesílání zpráv a POP nebo IMAP pro jejich příjem [10].

SMTP

Protokol používaný k odesílání zpráv a k jejich předávání mezi agenty MTA. SMTP je použit v případě, kdy MUA kontaktuje MTA a požaduje dodání zprávy, a také, když jeden MTA kontaktuje druhý MTA za účelem předání zprávy mezi nimi. V rozšířeních protokolu SMTP jsou též k dispozici prostředky pro ověřování uživatelů [10]. Komunikace protokolu probíhá ve formě čistého, nezašifrovaného textu, a to standardně na TCP portu 25 [3].

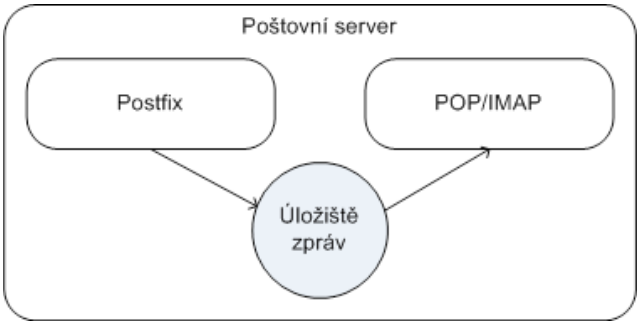
Norma RFC 2822, jenž v roce 2001 nahradila původní RFC 822, popisuje formát SMTP zprávy, která se skládá z hlavičky a těla zprávy, přičemž tělo může obsahovat přílohy s libovolným obsahem [12].

POP/IMAP

Pokud uživatel použije MUA a požaduje stažení zpráv, MUA se připojí k úložišti na serveru protokolem POP nebo IMAP a zprávy pro něj zajistí. Oba protokoly se zabývají výhradně převzetím doručených a uložených zpráv ze serveru, nikoliv jejich odesíláním.

Pomocí POP lze převzít všechny zprávy ze serveru a poté s nimi pracovat lokálně. Tím pádem má uživatel možnost pracovat s poštou, i když není právě připojen k síti. Mezi nevýhody protokolu POP patří problematické zacházení s poštou na více počítačích, špatná práce s více poštovními schránkami a nutnost stáhnout celou zprávu. Není tudíž možno stáhnout pouze předměty zpráv a na jejich základě se rozhodnout, zda danou zprávu uživatel vyžaduje. Komunikace probíhá standardně přes TCP port 110 [10].

IMAP umožňuje spravovat poštovní zprávy přímo na serveru. Tento protokol řeší výše jmenované problémy protokolu POP. V případě potřeby i IMAP dovoluje místní ukládání zpráv. Nevýhodou je zejména jeho značná složitost. Pro jeho potřeby je vyhrazen TCP port 143.



Obrázek 5 – Postfix a POP/IMAP [10]

Postfix uloží přijatou zprávu do úložiště zpráv. Na žádost uživatele protokoly POP nebo IMAP načítají zprávy z daného úložiště. Tuto situaci zobrazuje obrázek 5. Spolupráce mezi Postfixem a POP/IMAP tedy není přímá, musí se pouze dohodnout na formátu schránky a druhu zamykání. Postfix může též doručovat zprávy pomocí protokolu LMTP, více v [10].

Uživatelé, kteří mají přístup k prostředí unixového počítače, nepotřebují nutně přístup k úložišti zpráv pomocí POP/IMAP, jelikož mohou své MUA nakonfigurovat pro přímé čtení elektronické pošty z poštovního souboru na témže počítači [10].

3.1.3 Formát e-mailové zprávy

Zpráva elektronické pošty se skládá ze dvou částí: *záhlaví* a *těla*. Záhlaví je tvořeno položkami s názvy jako např. To (komu), From (od) nebo Subject (předmět) následovanými dvojtečkou (:) a obsahem dané položky. Jedinými povinnými položkami jsou Date (datum) a From. Tělo zprávy je od záhlaví odděleno prázdným řádkem a obsahuje samotný obsah zprávy. Formát těla je volný, ale mělo by obsahovat pouze ASCII znaky. V případě potřeby zaslání binárního souboru elektronickou poštou musí být i tyto soubory převedeny na znaky ASCII. Což je zajištěno kódováním MIME, více informací v [4].

3.1.4 Adresy obálky a záhlaví zpráv

Adresa uvedená v sekci To: v záhlaví zprávy nemá nic společného se skutečnou adresou, na kterou se má zpráva opravdu doručit. Skutečnou adresu doručení určuje tzv. adresa obálky. Sekce To: je z pohledu MTA pouhou součástí obsahu e-mailu. Typické chování agentů MUA je ovšem takové, že použijí adresu předanou v sekci To: jako adresu obálky, ale takové chování někdy nemusí být vyžadováno. Obálka musí obsahovat minimálně dvě položky. Jedná se o adresu odesílatele (MAIL FROM) a adresu příjemce (RCPT TO) [10].

Následující příklad formátu typické zprávy se záhlavím a tělem byl převzat a přepracován z [10].

```
Date: Mon, 8 Apr 2003 15:38:21 -0500
From: Costumer Service <info@oreilly.com>
To: <kdent@example.com>
Reply-To: <info@oreilly.com>
Message-ID: <01a4e2238200842@mail.oreilly.com>
Subject: Have you read RFC 2822?

Tady začíná tělo zprávy. Může mít další řádky, ale nemá.
```

3.1.5 DNS a MX záznamy

Poštovní server při přijetí zprávy prověřuje, zda se jedná o zprávu pro doménu, kterou sám obsluhuje. Pokud je tomu tak, je zpráva uložena do lokálního úložiště. V případě, že zpráva míří do jiné sítě, se server dotáže DNS na cílovou doménu. Důležitý je především tzv. MX záznam. Ten uvádí počítač, který se o poštu v dané doméně stará. Server tak může předat e-mail právě tomuto počítači, který zprávu buď lokálně uloží, nebo ji také přepošle dál. Podrobněji popsáno v [3].

3.2 Představení Postfixu

Postfix je poštovní server napsaný Wietsem Venemou [10], sloužící pro plnění základních úkolů při přenosu zpráv elektronické pošty. Hlavními úkoly, které Postfix provádí je přebírání zpráv od klientů, směrování, předávání zpráv jiným serverům a doručování do schránek [11]. Již na počátku vývoje byly stanoveny následující cíle, kterým se vývoj Postfixu musel podřídit [10].

Spolehlivost

Postfix detekuje případy, kdy se mnoho jiných softwarových systémů chová nepředvídatelně, jako je například nedostatek operační paměti či diskového prostoru, a snaží se všemi možnými způsoby zajistit stabilitu a spolehlivost systému.

Zabezpečení

Bezpečnostní politika Postfixu funguje na principu minimálních možných oprávnění pro každý proces, jelikož se předpokládá, že Postfix běží v nepřátelském prostředí.

Výkon

Jedním z hlavních požadavků je funkčnost při práci ve vysokém zatížení. Speciálními technikami zajišťuje, že jeho rychlost nezahltí jiné systémy. Jedná se například o limitování počtu nových procesů, které je nutno vytvořit.

Flexibilita

Postfix je tvořen různými programy a subsystémy. Tímto přístupem nabízí vysokou flexibilitu, přičemž všechny části lze snadno nastavovat pomocí konfiguračních souborů.

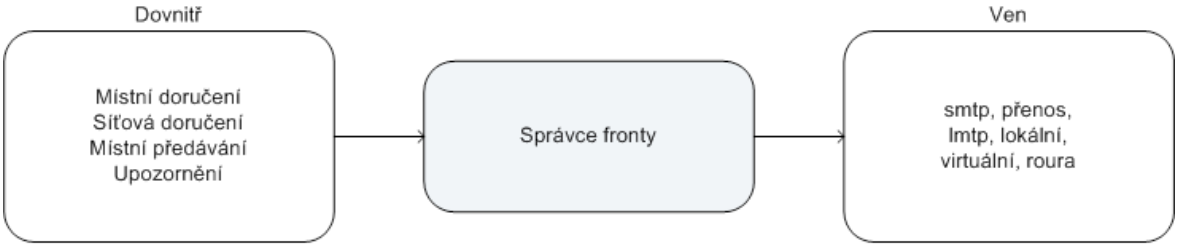
Kompatibilita se Sendmailem

Sendmail je stále používaný poštovní server, který může být díky zpětné kompatibilitě nahrazen Postfixem. Hlavní nevýhodou Sendmailu je jeho monolitická architektura, která je zdrojem bezpečnostních rizik.

3.3 Architektura systému Postfix

Postfix je charakteristický svým modulárním návrhem. Hlavní démon se nazývá *master*, ten volá a řídí výkonné moduly za účelem realizace konkrétních úkolů. Tyto procesy mohou být ukončeny buď v případě splnění úlohy nebo po vypršení časového limitu [10]. V případě neočekávaného ukončení výkonného modulu jej *master* restartuje. V případě, že se problém u modulu opakuje, čeká hlavní démon po stanovenou dobu (implicitní hodnota je jedna minuta), aby nebyl systém zbytečně zatěžován neúspěšnými restarty.

Modul *master* dokáže spravovat i komponenty, které nejsou součástí systému Postfix. Taková situace nastává typicky, pokud se o finální doručení zprávy do schránky stará nějaký agent MDA [11].



Obrázek 6 – Celkový přehled architektury Postfixu [10]

Na obrázku 6 je zevrubně znázorněna architektura Postfixu. Ten přijímá zprávy, řadí je do fronty a poté zavolá příslušný MDA, který zprávu nakonec doručí [10].

3.3.1 Postfix jako MTA

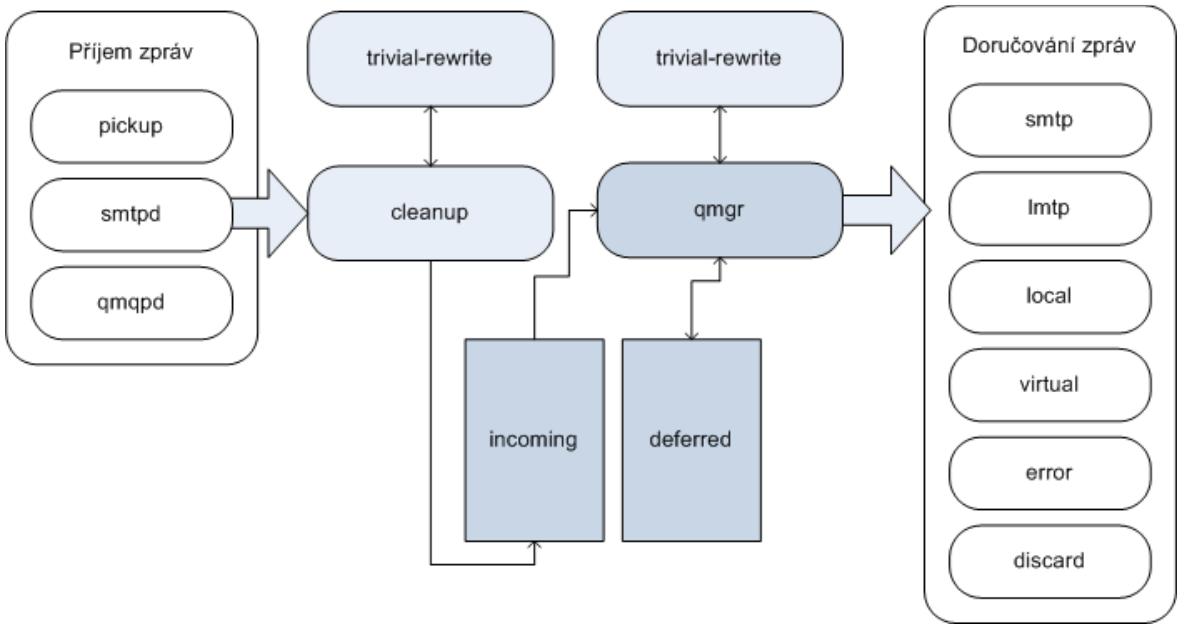
Zpráva elektronické pošty může být do systému přijata několika cestami. Typicky je přijata přes síťovou SMTP relaci nebo přes rozhraní sendmail[3] u lokálně odesílaných zpráv. V případě síťové relace zprávu přijímá modul *smtpd.* Ihned na vstupu je zpráva zpracována modulem *cleanup.* Ten se postará o opravu a doplnění hlaviček, v případě potřeby provede expanzi cílových obálkových adres. K tomuto účelu může použít modul *trivial-rewrite.* Následně vloží zprávu do fronty příchozích zpráv označovanou jako *icoming* a upozorní správce fronty (*qmgr*), že se ve frontě nachází nová zpráva.

Správce fronty udržuje tzv. „aktivní frontu“, která je jakýmsi pohledem do skutečné fronty příchozích a odložených zpráv. Tyto zprávy se pokouší předat dál. Pokud je pokus úspěšný, zpráva je z fronty odstraněna. V případě neúspěchu je zpráva zařazena do fronty odložených zpráv (*deferred*) a po určitém čase se opět aktivuje pro další doručovací pokus.

Dále se o zprávu starají doručovací agenti. Modul *local* doručí zprávu lokálním uživatelům, modul *virtual* pak virtuálním. Moduly *smtp* a *lmtp* předávají zprávu dalšímu SMTP serveru.

V případě, že nelze zprávu z nějakého důvodu doručit, Postfix použije modul *discard*, který zprávu jednoduše zahodí a modul *error*, který ji vrátí odesílateli s dočasnou chybou *retry* nebo trvalou chybou *error* [11].

[3] Funkcionalitu zde zajišťuje Postfix, nejedná se o poštovní server Sendmail.



Obrázek 7 – Průchod zpráv systémem Postfix [11]

3.4 Zabezpečení Postfixu

Emailové servery jsou již od samé jejich podstaty vystaveny příjmu zpráv z nedůvěryhodných zdrojů. Operují v nepřátelském prostředí a tomu musí být přizpůsobeno jejich zabezpečení. Postfix uplatňuje tzv. vícevrstvý přístup a modulární návrh k řešení jeho ochrany. Právě modulární architektura je klíčová pro bezpečnost celého poštovního systému. Každý proces Postfixu je spuštěn s co možná nejnižším oprávněním nezbytným pro provedení dané úlohy. Procesy, které nejsou pro systém nezbytně nutné je možno vypnout. Tím pádem nemohou být zneužity potencionálním útočníkem. Procesy jsou též vzájemně izolovány, tudíž využívají meziprocesní komunikaci v nejmenší možné míře.

Většinu procesů Postfixu má pod kontrolou důvěryhodný řídící démon. Tyto procesy neběží jako uživatelské podřízené procesy, a proto jsou imunní vůči potížím vyplývajícím z dědičných vztahů nadřazený-podřazený.

Unixové systémy mají schopnost změnit kořenový adresář aplikací (*chroot*), tím pádem může být adresář Postfixu nastaven jako kořenový. Když pak Postfix běží, jeho pohled na systém souborů je omezen jen na strom právě pod jeho adresářem a nic nad touto úrovní nemůže pozorovat. To znamená, že v případě úspěšeného napadení poštovního serveru nemá útočník přístup ke kritickým systémovým adresářům, které obsahují citlivé informace o celém serveru.

Ani útoky typu DDoS[4] nejsou vůči poštovnímu serveru moc efektivní. Postfix je koncipován pro pro provoz pod vysokým zatížením. V případě, že systému dojde volná

[4] Distributed Denial of Service, technika útoku přehlcením serveru požadavky.

operační paměť či diskový prostor, upustí od operace, kterou se snažil učinit, a dá tak systému možnost vzpamatovat se.

Postfix je koncipován tak, aby si dokázal poradit s nejrůznějšími útoky a nepříznivými podmínkami. Toho je dosaženo vestavěnou robustností. Základní tezí autora Postfixu, Wietse Venemy je, že se zajímá o vytvoření softwaru, který funguje požadovaným způsobem bez ohledu na okolní podmínky, více než o samotné jeho zabezpečení [10].

3.5 Instalace

Zdrojové kódy Postfixu lze získat z jeho domovských stránek[5] a poté je stačí přeložit pro danou platformu. Je však pravděpodobné, že se zde již nachází předkompilovaný balíček pro použitou platformu [10].

Instalace Postfixu je v tomto případě provedena standardními nástroji balíčkovacího systému APT, kterým Debian GNU/Linux disponuje. V kombinaci s neuvěřitelně objemnými repozitáři[6] představuje silný nástroj pro instalaci jakéhokoliv softwaru.

Postfix se vyskytuje v základních repozitářích Debianu, stačí tedy v otevřeném příkazovém interpretu zadat jako uživatel *root* příkaz:

```
# aptitude install postfix
```

Tím proběhne instalace poštovního serveru Postfix i se všemi případnými závislostmi a současně odebrání poštovního serveru *exim* [13].

3.6 Konfigurace

Postfix se po instalaci nachází ve výchozí konfiguraci. To znamená, že je schopen přijímat a odesílat poštu, kterou obdrží. Základní nastavení nemusí být vždy vhodné a proto je v drtivé většině případů nutno poštovní server nakonfigurovat [14].

Hlavními konfiguračními soubory Postfixu jsou *master.cf* a *main.cf*. Skoro veškerá nastavení se realizují prostřednictvím souboru *main.cf*, který je typicky umístěn v adresáři */etc/postfix*. Hlavní konfigurační soubor je možno měnit příkazem *postconf* nebo přímou editací v libovolném textovém editoru [10].

Formát konfiguračního souboru *main.cf* vypadá následovně:

```
volba = hodnota
```

Parametry je možno zapisovat v libovolném pořadí. Více parametrů lze oddělit mezerou, čárkou, tabulátorem nebo novým řádkem. Komentářům předchází znak # [10].

[5] http://www.postfix.org/
[6] Server, který centrálně spravuje předkompilované balíčky pro linuxové distribuce.

3.6.1 První spuštění

Po instalaci Postfixu je nutno zjistit plně kvalifikovaný název hostitele, ten se skládá jak z názvu hostitele, tak z názvu domény, ve které se hostitel nachází.

Název hostitele zjistíme unixovým příkazem:

```
$ hostname
```

Je-li výsledkem příkazu pouze název hostitele, nedokáže Postfix stanovit plně kvalifikovaný název, který je nutný pro správnou identifikaci systému v síti [10].

Název domény změníme příkazem:

```
# postconf –e "myorigin = bp.cz"
```

Formát elektronické adresy bude tedy uživatel@bp.cz.

Zjištěný název hostitele použijeme a vytvoříme tak plně kvalifikovaný název:

```
# postconf –e "myhostname = server.bp.cz"
```

Pokud tento příkaz nezadáme, Postfix standardně použije název hostitele poskytnutý samotným systémem [10].

Posledním klíčovým krokem při prvním spuštění je nastavení domén, kterým bude instalovaný systém přeposílat zprávy. Tím, že žádné takové domény neuvedeme, zajistíme, že poštovní server nebude sloužit pro přenos zpráv do nedůvěryhodných sítí [15].

```
# postconf –e "relay_domains ="
```

Provedeme restart poštovního serveru:

```
# postfix reload
```

Nyní je Postfix funkční a připraven pro další nastavení.

3.6.2 Nastavení pro LAN

Poštovní server bude k dispozici všem klientům na místní síti. Proto je nutno provést několik dalších nastavení.

Parametr *mydestination* určuje, které domény bude poštovní server obsluhovat lokálně a které musí pro úspěšné doručení předat jinému stroji. Pro nastavení serveru, který dokáže obsloužit celou síť LAN, musíme zadat následující:

```
# postconf –e "mydestination = $myhostname localhost.$myorigin localhost
$myorigin"
```

Pro vyhnutí se smyčkám při doručování je nutno zadat všechna jména serveru, včetně *$myhostname* a *localhost.$myorigin* [16].

Postfix ve výchozím nastavení přeposílá zprávy všem klientům v autorizovaných sítích, které jsou uvedeny pod parametrem *mynetworks*. My tuto volbu potvrdíme následujícím příkazem, který říká, že Postfix bude přeposílat poštu všem SMTP klientům na stejné podsíti jako se nachází samotný server. Samotnou informaci o podsíti získá Postfix z výstupu unixového příkazu *ifconfig*, který slouží pro správu síťových rozhraní [16].

```
# postconf –e "mynetworks_style = subnet"
```

Parametr *relayhost* určuje další server, který bude následovat v cestě doručení zprávy, pokud se jedná o mail mimo lokální doménu. V našem případě bude veškeré doručovaní pošty fungovat v rámci LAN a tudíž necháme tento parametr prázdný [16].

```
# postconf –e "relayhost ="
```

Parametr *inet_interfaces* určuje na který síťových rozhraních poštovní server naslouchá. Potvrdíme výchozí nastavení, při kterém Postfix naslouchá na všech aktivních rozhraních.

```
# postconf –e "inet_interfaces = all"
```

Poslední nastavení se týká samotného protokolu IP. Můžeme se zvolit, zda Postfix akceptuje IPv4, IPv6 nebo oboje najednou. Možnými parametry jsou: *ipv4*, *ipv6* a *all.*

```
# postconf –e "inet_protocols = ipv4"
```

Poté je již Postfix připraven pro fungování na lokální síti.

3.6.3 Aliasy

Skutečné poštovní servery mají často zavedeny adresy jako *admin* nebo *info*, tito uživatelé ovšem v systému nemusí vůbec existovat. Pro doručení zprávy skutečnému uživateli slouží tzv. *alias.* K tomuto účelu slouží soubor */etc/aliases*, který lze otevřít v běžném textovém editoru [3].

Jeho formát je jednoduchý:

```
alias:      uživatel
```

Uvedeme tedy alias, který chceme používat a následně ho přesměrujeme na existujícího uživatele systému.

Po každé editaci souboru s aliasy je nutno provést příkaz, který aktivuje příslušné změny:

```
# postalias /etc/aliases
```

3.6.4 Courier

Pro přístup k poště uložené na serveru lze používat protokol POP nebo IMAP. Výhody a nevýhody jednotlivých protokolů již byly zmíněny výše. Programů, které dokáží poskytnout tento přístup je mnoho. V případě našeho serveru použijeme balík Courier, který poskytuje šifrovanou i nešifrovanou variantu obou protokolů [3].

Nainstalujeme balíček *courier-imap* pro přístup k poště přes moderní protokol IMAP:

```
# aptitude install courier-imap gamin
```

Courier ukládá přijaté zprávy do složky Maildir v domácím adresáři uživatele. Nejprve tedy upravíme soubor */etc/skel*, který slouží jako kostra domovského adresáře pro každého nově vytvořeného uživatele systému [17].

```
# maildirmake /etc/skel/Maildir
```

Pro již existující uživatele musí být složka vytvořena explicitně:

```
# cp -r /etc/skel/Maildir /home/uzivatel
```

Dále je nutno změnit vlastníka této složky na příslušného uživatele. Pokud by ji vlastnil jiný uživatel, doručovaní do schránky by nefungovalo.

```
# chown -R uzivatel:uzivatel_skupina /home/uzivatel/Maildir
```

Posledním krokem je zavedení složky Maildir do Postfixu:

```
# postconf -e "home_mailbox = Maildir/"
```

Konfigurační soubory Couriera se typicky nacházejí v adresáři */etc/courier*, kde je možno modifikovat jeho nastavení. Obecně však není potřeba měnit nic a přístup k poště přes protokol IMAP funguje ihned po instalaci a vytvoření složky pro uživatelské zprávy [3].

4 Souborový server – Samba

4.1 Co je to Samba?

Samba je sada velmi užitečných síťových nástrojů, která umožňuje unixovým systémům využívat protokol SMB (Server Message Block). Tento protokol používá pro přenos dat v síti řada operačních systémů, včetně MS Windows či OS/2. Autorem Samby je Andrew Tridgell, který stále aktivně řídí vývojový tým dobrovolníků. Díky protokolu SMB je umožněno serverům s unixovými operačními systémy fungovat jako server pro stanice se systémy Windows a poskytovat například následující služby [18]:

- sdílení struktur adresářů a struktur systému souborů,
- sdílení tiskáren,
- možnost prohlížení síťového okolí,
- autentizace klientů.

Samba též disponuje nástrojem, jenž umožňuje uživatelům unixových systémů přístup k sdíleným prostředkům, které na síti nabízí systémy Windows.

Pro svůj běh využívá Samba dvou důležitých démonů, kteří poskytují sdílené prostředky klientům [18]:

- *smbd* – řídí sdílení souborů a tiskáren a umožňuje autentizaci a autorizaci SMB klientů. Komunikace funguje na TCP portu 139 nebo 445 [19].

- *nmbd* – umožňuje překlad jmen protokolu NetBIOS. Též podporuje procházení sítí. Pro svoje potřeby používá UDP porty 137 a 138 [19].

Samba obsahuje také několik pomocných programů. Patří mezi ně například *smbclient*, který umožňuje pracovat s SMB svazky podobně jako s FTP serverem, *smbmount,* sloužící pro připojování SMB svazků jako lokálních disků, a *smbpasswd* pro správu hesel [19].

V roce 2007 dala firma Microsoft k dispozici kompletní dokumentaci k protokolu SMB, tím umožnila kvalitnější a snazší práci vývojářům projektu Samba [20]. Sám Microsoft používá pro SMB jinou zkratku, a to CIFS (Common Internet File System) [18].

4.1.1 NetBIOS

Jelikož byl protokol SMB navržen pro sítě bez protokolu TCP/IP, obsahoval prostředky pro identifikaci počítačů nezávisle na běžných TCP/IP službách jako je například DNS. K tomuto účelu sloužilo softwarové rozhraní NetBIOS [19]. Klientům poskytuje jména, která slouží k jejich jednoznačné identifikaci v rámci sítě. Každý klient může mít několik jmen, pokud nejsou zaregistrována jiným klientem, přičemž délka jména je omezena na 15 znaků [18]. Vrstva NetBIOS byla původně protokolem SMB vyžadována, to už však není nutné a je používán přímo protokol TCP/IP [21]. Více informací o NetBIOS a WINS,

který sloužil jako jmenný server pro systémy Windows a byl zavržen od verze Windows 2000, lze dohledat v [18].

4.1.2 Domény a pracovní skupiny

Domény a pracovní skupiny definují různé uspořádání počítačů v síti. Hlavní rozdíl představuje způsob správy jednotlivých stanic. Všechny počítače se systémy MS Windows musí být součástí domény nebo pracovní skupiny [22].

Doména

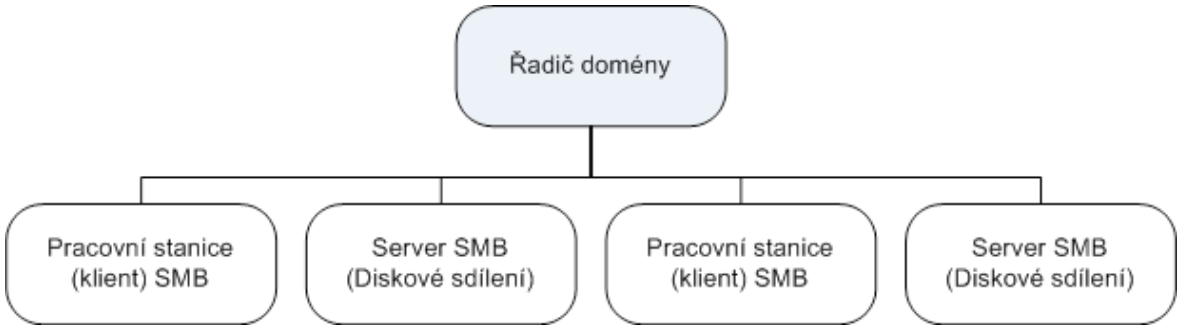
Jedna či více stanic v síti vystupují v roli serveru. Tím je umožněno správci sítě řídit zabezpečení a oprávnění všech klientů v doméně. Změny, které správce provede, se projeví na všech počítačích v doméně. Jestliže má uživatel účet v doméně, může se pod ním přihlásit na jakémkoliv počítači v doméně. Domény mohou obsahovat tisíce počítačů a ty mohou být v různých sítích [22].

Pracovní skupina

V pracovní skupině jsou si počítače rovnocenné. Každý počítač má svojí vlastní sadu uživatelských účtů, není tedy možné, aby uživatel mohl přistupovat ke svému účtu na libovolné stanici v síti. Skupina obvykle nemá více než dvacet stanic a všechny stanice musí být ve stejné síti či podsíti [22].

4.1.3 Řadiče domény

Hlavní úlohou řadiče domény je správa uživatelských účtů. Jednou z hlavních funkcí je autentizace uživatelů. Autentizace je proces ověření identity uživatele, většinou na základě hesla. Každý řadič domény NT používá standardně *správce zabezpečení účtů* (SAM), kde jsou uložena uživatelská jména a hesla. Více o bezpečnostním modelu řadiče domény v [18].



Obrázek 8 – Jednoduchá doména Windows [18]

Primární řadič domény (PDC)

Aktivní řadič se nazývá řadičem primárním. Jedná se o počítač v síti, který řídí doménu. Obsahuje seznam všech uživatelů a pracovních stanic. Poskytuje záložnímu řadiči domény svá data pro synchronizaci.

Záložní řadič domény (BDC)

Počítač, který přebírá povinnosti PDC v případě jeho nedostupnosti. BDC často provádí synchronizaci dat SAM s PDC. Tato data má BDC k dispozici jako kopie pouze pro čtení. Může tedy svá data aktualizovat pouze s PDC [18].

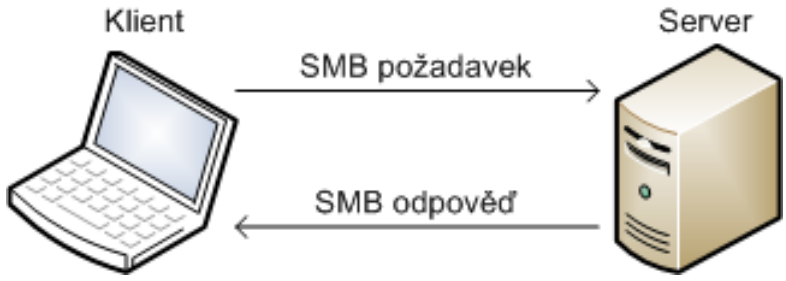
4.1.4 Samba vs. NFS

Síťový souborový systém NFS poskytuje sdílení souborů mezi unixovými systémy. I pro systémy Windows existuje několik klientů, kteří jsou schopni zajistit sdílení mezi oběma platformami, ovšem je jednodušší jednou nainstalovat Sambu na server, a tím zajistit pro klientské stanice se systémem Windows přirozené prostředí pro sdílení, než na každý počítač v sítí instalovat NFS klienta. Oproti Sambě disponuje pouze sdílením souborů, nenabízí služby pro autentizaci a řízení přístupu, ale přesto je často na čistě unixových sítích používán pro jeho bezproblémovou správu a podporu [3].

4.2 Protokol SMB

SMB je síťový protokol aplikační vrstvy[7], který zahrnuje operace pro manipulaci se soubory a tiskárnami včetně otevírání a zavírání souborů, vytváření a odstraňování souborů či složek a řízení tiskové fronty tiskárny. Každou takovou operaci lze zakódovat do zprávy SMB a přenést ji na server či ze serveru [18].

Protokol SMB funguje jako protokol typu *požadavek-odpověď*. To znamená, že klient zasílá požadavek, server ho přijme, zpracuje, prověří přístupová práva klienta vzhledem ke sdílenému prostředku a na jejich základě zašle odpověď ve formě bloku SMB. Sdílené prostředky poskytované serverem jsou identifikovány univerzální síťovou adresou UNC[8] [19].



Obrázek 9 – Přístup požadavek-odpověď [18]

Strukturu zprávy SMB lze rozdělit na záhlaví s pevnou délkou a příkaz, jehož délka závisí na obsahu zprávy. Více o struktuře SMB zprávy v [18].

[7] Z pohledu modelu ISO/OSI
[8] UNC je ve tvaru \\jmeno_serveru\jmeno_sdileneho_prostredku

4.2.1 Řízení přístupu

Pro přístup ke sdíleným prostředkům jsou typicky využívány dva přístupy. Jedná se řízení přístupu na úrovni sdíleného prostředku a řízení přístupu na uživatelské úrovni [23].

Řízení přístupu na úrovni sdíleného prostředku (share)

Každý sdílený prostředek má svoje heslo a klientům v síti je k němu dovoleno přistupovat na základě jeho znalosti. Po úspěšném zadání hesla je klientovi přiřazen identifikátor NID (Network ID). Pro jeden sdílený prostředek může být nastaveno několik hesel, každé s jinou úrovní přístupu [23].

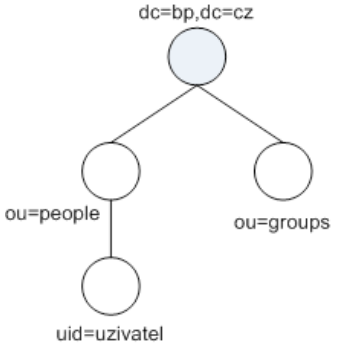
Řízení přístupu na uživatelské úrovni (user)

Server požaduje při přihlášení uživatele jeho jméno a heslo. Jsou-li údaje zadány správně, je uživateli přiřazen identifikátor UID (User ID). Na jeho základě server přiděluje přístupová práva ke sdíleným prostředkům. K rozlišení jednotlivých procesů stejného uživatele slouží identifikátory PID (Process ID) a MID (Multiple ID) [23].

4.3 LDAP

LDAP je jednoduchý protokol pro přístup k adresářovým službám. Pod pojmem adresářová služba si lze představit databázi, do které se málo zapisuje, ale často se z ní čte. Do LDAP lze uložit informace o účtech pro Windows i unixové systémy, kontakty na zaměstnance a mnoho dalších užitečných informací. Velkou sílu má LDAP při nasazení v roli síťové autentizační databáze, především díky již zmíněné možnosti spravovat účty unixových systémů i systémů Windows [19].

LDAP organizuje data do stromové struktury. Pro identifikaci objektů se používá jednoznačných identifikátor DN (Distinguished Name), který uvádí pozici záznamu ve stromě. K označení jmenného kontextu záznamu slouží DC (Domain Component). Identifikátor OU se používá například k rozlišení různých oddělení jedné organizace. Pro výměnu LDAP dat se používá formát LDIF [19].



Obrázek 10 – Stromová struktura záznamů v LDAP [24]

V případě použití LDAP jako síťové autentizační databáze v doméně NT, nahradí standardní SAM. Více o protokolu LDAP k dispozici v [19].

4.4 Kerberos

Kerberos je autentizační systém, který je ideální pro autentizaci různých systémů pomocí různých protokolů. Hlavním cílem Kerbera je poskytnout centrální autentizaci klientů, ale funguje i opačným směrem, tudíž dokáže klienty ujistit o identitě klientem požadovaných služeb. Největší výhodou Kerbera je systém jediného přihlášení (SSO). Uživatel se tak může přihlásit ke svému počítači a může využívat služeb dalších serveru v síti bez nutnosti zadávat ke každé službě heslo zvlášť. Těmito službami může být například FTP a SSH server, přístup k poště protokolem IMAP, atd [19].

Základním pojmem Kerbera je *principál*. Principál je řetězec, který představuje identitu uživatele. Nemusí se však jednat pouze o fyzickou osobu. Vlastní principál mají i počítače a kerberizované služby. Principál má tvar *primary/instance@REALM*. První část, *primary*, představuje buď uživatelské jméno nebo jméno služby. Část *instance* má pro každý typ principálu jiný význam. Pokud se jedná o principál uživatele, je instance rozšiřující částí, která specifikuje účel uživatele v systému. V případě principálu počítače obsahuje část instance jeho plně kvalifikované doménové jméno. Poslední část *REALM* (říše) představuje oblast spravovanou Kerberos serverem. Jedná se o analogii s doménou ve světě systémů Windows. Říše je citlivá na velikost písmen a je vždy zapsána písmeny velkými [24].

Autentizace v systému Kerberos je prováděna pomocí tzv. *lístků*. Jedná se o zašifrovaný blok dat, pomocí kterého se systémy navzájem autentizují. Lístky Kerbera jsou poskytovány vždy na určitou, předem definovanou, dobu. Z toho vyplývá, že v síti, kde je autentizace uživatelů zajištěna systémem Kerberos, je nutné synchronizovat čas všech stanic a serverů v síti. Proto je prakticky nutné zajistit časovou synchronizaci pomocí serveru NTP. Více o způsobu šifrovaní a komunikace serveru s klientem v [19] a [24].

4.5 Instalace

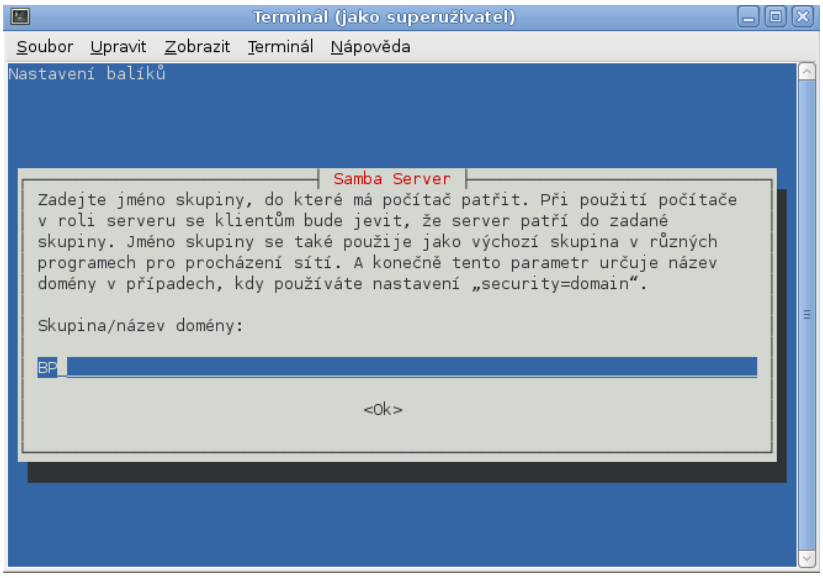
Stejně jako v případě Postfixu, i Sambu je možno přeložit a instalovat přímo ze zdrojových kódů dostupných na domovské stránce[9] Samby nebo instalovat pomocí nástroje *aptitude* přímo ze základních repozitářů Debianu.

V příkazovém interpretu tedy zadáme jako uživatel *root* tento příkaz:

```
# aptitude install samba smbfs smbclient samba-doc
```

Následně zvolíme, v jaké skupině se náš server nachází.

[9] http://www.samba.org/



Obrázek 11 – Volba jména skupiny Samby

4.6 Konfigurace

Hlavním konfiguračním souborem Samby je */etc/samba/smb.conf*. Syntaxe jeho zápisu je totožná s konfiguračním souborem Postfixu */etc/postfix/main.cf*, která je popsána výše. Výjímku zde tvoří komentáře, které jsou uvozené středníkem. Pokud se daná volba nachází právě za středníkem, znamená to, že jejím povolením se Samba bude lišit od jejího výchozího chování [18].

Soubor smb.conf se skládá ze tří hlavních sekcí. První je sekce globálního nastavení označena jako *[global]*. Každý sdílený prostředek má také svoji vlastní sekci. Dvě vyhrazené sekce jsou *[homes]* a *[printers]*.

V celém konfiguračním souboru lze používat následující proměnné, za které pak bude doplněn příslušný obsah [3]:

- %v – verze Samby,
- %T – čas a datum,
- %m – jméno klienta,
- %h – jméno stroje,
- %M – jméno stanice,
- %D – pracovní skupina uživatele,
- %H – domovský adresář uživatele.

Po každé modifikaci souboru *smb.conf* je doporučeno spustit příkaz *testparm* pro ověření správného syntaktického zápisu souboru:

```
$ testparm
```

Záložní kopie konfiguračního souboru *smb.conf* se nachází v adresáři */usr/share/samba*.

4.6.1 Globální sekce

Sekce *[global]* se stará o globální nastavení serveru. To ovlivňuje chování všech poskytovaných služeb. Následuje výčet nejdůležitějších direktiv. Tato sekce bude v dalších krocích konfigurace měněna.

```
[global]
  workgroup = NASE_DOMENA
  server string = %h server
  dns proxy = no
  unix password sync = yes
  security = user
```

Volbou *workgroup* dáváme najevo, do jaké pracovní skupiny či domény Samba server patří.

```
workgroup = NASE_DOMENA
```

Hodnotou nastavenou ve volbě *server string* se bude server prezentovat při prohlížení sítě.

```
server string = %h server
```

Direktiva *dns proxy* říká, zda bude *nmbd* hledat NetBIOS jména přes systém DNS. Ve výchozím stavu je tato možnost znemožněna:

```
dns proxy = no
```

Pro udržení synchronizace hesel systému a Samby samotné se používá direktiva *unix password sync*. Kdykoliv uživatel změní své Samba heslo, bude vyzván ke změně svého hesla do systému samotného.

```
unix password sync = yes
```

Poslední, a velmi důležitou volbou je *security*. Ta slouží pro nastavení řízení přístupu k sdíleným prostředkům.

```
security = user
```

4.6.2 Uživatelé

Samba ve výchozím stavu nešifruje hesla. Toto nastavení změníme a server s klientem tak budou uchovávat hesla v zašifrované podobě. Ověření probíhá tak, že server zašle klientovi výzvu, kterou klient použije pro vygenerování hashe zašifrovaného hesla. Hash je pak odeslán serveru, který provede obdobnou operaci a výsledky porovná.

Samba akceptuje šifrovaná hesla po přidání následující direktivy do sekce *[global]*:

```
encrypt passwords = yes
```

Jelikož Sambě není umožněno ověřovat heslo uživatele systému ze standardních souborů */etc/passwd* a */etc/shadow*, musí být heslo pro každého uživatele systému, který chce mít k dispozici služby souborového serveru, vytvořeno odděleně pomocí nástroje *smbpasswd* [25].

To znamená, že každý uživatel Samby musí mít nutně vytvořen účet v samotném systému, pokud Samba nevyužívá služeb libovolné centrální databázové služby.

Přidání nového uživatele se jménem *jmeno_uzivatele* do Samby:

```
# smbpasswd –a jmeno_uzivatele
```

Odstranění uživatele jmeno_uzivatele:

```
# smbpasswd –x jmeno_uzivatele
```

Uživatel si může změnit své heslo tímto příkazem [18]:

```
$ smbpasswd
```

4.6.3 Sdílení adresářů

Domovské adresáře

Sdílení domácích adresářů uživatelů je natolik důležitě, že pro něj Samba definuje speciální sekci *[homes]*. Výchozím adresářem pro sdílení je domovský adresář uživatele uvedený v souboru */etc/passwd*. Na sítí je potom k dispozici s názvem podle jména uživatele [19].

Jako definici sekce *[homes]* použijeme výchozí nastavení Samby:

```
[homes]
  comment = Home Folder
  browseable = no
  read only = no
  create mode = 0750
  valid users = %S
  public = no
```

Direktiva *comment* určuje hodnotu, kterou uživatelé uvidí vedle jména svazku při prohlížení sítě.

```
comment = Domaci adresar
```

Voba *browseable* říká, zda je sdílený svazek viditelný při prohlížení sítě anebo je přístupný pouze přes jeho název. V případě, že nastavíme *browseable = yes*, objeví se uživatelům navíc adresář *HOMES*.

```
browseable = no
```

Výchozí nastavení Samby nechává sdílené svazky v režimu pouze pro čtení. To znamená, že direktiva *read only* je ve výchozím stavu nastavena na *yes*. Nastavíme tedy možnost zápisu do domovských adresářů pomocí *read only = no*.

```
read only = no
```

Volba *create mode* je synonymem *create mask*. Těmi nastavíme maximální přípustné oprávnění nových souborů, resp. složek u direktivy *directory mask*. To znamená, že nově vytvořené soubory (složky u *directory mask*) obdrží zde nastavená práva. Pro jejich sdělení se používá tradiční unixový oktalový zápis práv, kterému předchází znak nula (0). Více o právech v [4].

```
create mode = 0750
```

Direktivy *invalid users* a *valid user*s stanovují, kteří uživatele mají přístup do svazku zakázán, respektive povolen. Přístup lze zakázat či povolit i celým unixovým skupinám pomocí zavináče (@) nebo znaku plus (+), například:

```
valid users = %S
```

Každý sdílený svazek může být označen direktivou *public*, která nabývá hodnot *yes* nebo *no*. Při nastavení *public* = *yes* není pro přístup ke sdílenému svazku třeba heslo.

```
public = no
```

Obecné sdílené adresáře

Uživatelům sítě můžeme poskytovat různé sdílené svazky. To může být na mnoha serverech jediný účel Samby. Tyto svazky mohou obsahovat nejrůznější data a mohou být poskytovány s rozličnými právy.

Vytvoříme tedy následující sdílený svazek:

```
[sdileny_adresar]
  comment = Sdilena data uzivatelu
  path = /home/samba/sdileny_adresar
  available = yes
  valid users = @users
  read only = no
  read list = zly_uzivatel
  force user = ucet_vedouciho
  create mask = 0660
  directory mask = 0770
```

Pro jednoduché vytvoření sdíleného adresáře stačí přidat do konfiguračního souboru jeho název v hranatých závorkách:

```
[sdileny_adresar]
```

Už při tomto nastavení je adresář uživatelům k dispozici a bude odpovídat adresáři */tmp* na serveru. Je zřejmé, že toto nastavení nebude optimální, a proto existuje direktiva *path*, která připojí do sdíleného svazku libovolný adresář na serveru:

```
path = /var/lib/samba/sdileny_adresar
```

Direktiva *available* může být nastavena na hodnoty *yes* nebo *no* a určuje, zda je sdílený svazek přístupný. Při nastavení na *available = no* se Samba chová, jako kdyby sdílený svazek neexistoval.

```
available = yes
```

Pomocí *write list* můžeme specifikovat uživatele, kteří budou mít právo zapisovat do sdíleného adresáře i pokud bude připojen pouze pro čtení. Direktiva *read list* naopak určuje uživatele, kteří nemohou na svazek zapisovat [19].

```
read list = zly_uzivatel
```

Nastavení vlastníka svazku lze provést volbou *force user*. Samba se bude za nastaveného uživatele vydávat při každém přístupu k sdílenému adresáři. Pokud uživatele vytvoří nějaký soubor, bude jako jeho vlastník uveden nastavený parametr:

```
force user = ucet_vedouciho
```

V případě, že tuto direktivu nenastavíme, Samba použije uživatelské jméno poskytnuté klientem.

Toto je pouze základní přehled možných direktiv pro nastavení sdílených svazků, celkový přehled je k dispozici v [18].

4.6.4 Sdílení tiskáren

Před samotným sdílením tiskáren pomocí Samby musí být nainstalován tiskový systém, který se stará o správu tiskové fronty a úloh. Standardně je v Debianu 6.0 používán systém CUPS, který umožňuje administraci pomocí přehledného webového rozhraní.

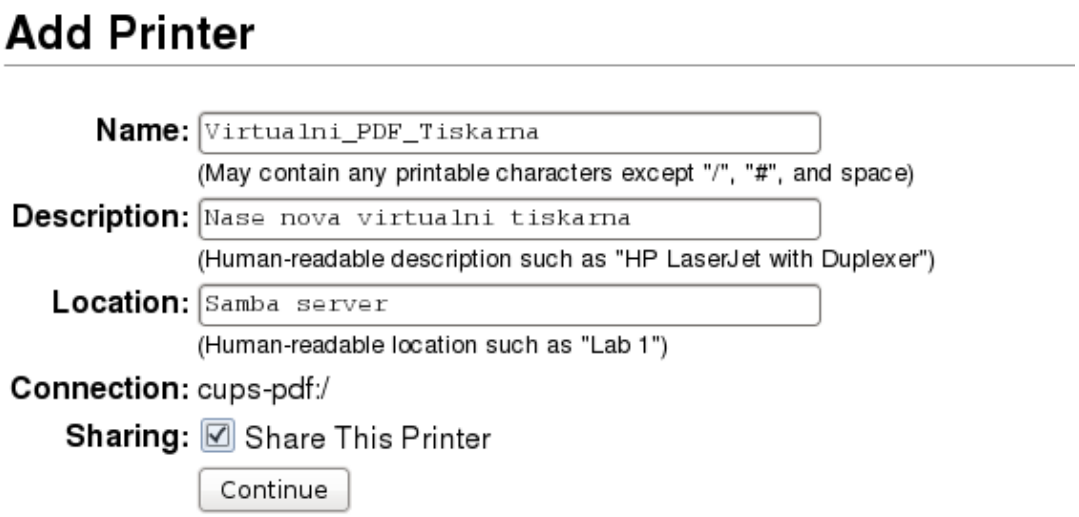
Pro sdílení tiskárny potom stačí definovat v systému CUPS tiskárnu, jenž je fyzicky připojena k serveru, a povolit sdílení všech tiskáren, které jsou k serveru připojeny. Nejsnazším způsobem, jak definovat novou tiskárnu, je přístup přes webové rozhraní pod adresou:

```
http://localhost:631
```

Zde v sekci *Administration* povolíme volbu „*Share printers connected to this system*“, tím zajistíme síťové sdílení všech lokálně připojených tiskáren, a následně již můžeme přidat tiskárnu volbou „*Add Printer*“. Vyplníme přihlašovací údaje uživatele *root* a můžeme přidat libovolnou tiskárnu. V tomto případě bude sdílená tiskárna pouze virtuální a bude poskytovat vytváření PDF souborů, které se uloží do domácího adresáře uživatele. Pro možnost vytvoření PDF tiskárny je nutno nainstalovat balíček *cups-pdf* [26]:

```
# aptitude install cups-pdf
```

Následně vyplníme sekci „*Add Printer*“:



Obrázek 12 – Přidání tiskárny v systému CUPS

V následujícím kroku musíme zvolit ovladače tiskárny. V našem případě vybereme možnost *Generic* a *Generic CUPS-PDF Printer*. Při přidávání opravdové tiskárny vybereme ovladače vhodné pro naši tiskárnu, nebo lze předat pouze soubor s popisem tiskárny, PPD, který získáme od výrobce.

Nyní je tiskárna plně funkční v rámci lokálního použití na serveru. Pro síťový tisk musíme nastavit v sekci *[global]* souboru *smb.conf* následující:

```
printing = CUPS
printcap name = CUPS
load printers = yes
```

Tím zajistíme užití systému CUPS jako výchozího tiskového systému a volbou *load printers = yes* řekneme Sambě, ať automaticky předává klientům informace o dostupných tiskárnách [19].

Dalším krokem je vytvoření oddílu *[printers]*, zde necháme direktivy ve výchozím nastavení:

```
[printers]
   comment = All Printers
   browseable = no
   path = /var/spool/samba
   printable = yes
   guest ok = no
   read only = yes
   create mask = 0700
```

Samba používá pro distribuci ovladačů klientům svazek *[print$]*. Ten se hodí zejména v rozsáhlých sítích, kde je pro administrátora neúnosné instalovat ovladače lokálně na každém klientském počítači. Více o tomto svazku v [19].

Po těchto krocích a nutném restartu Samby je tiskárna připravena pro síťový tisk.

4.6.5 Samba + Kerberos + LDAP

Předpokladem pro správné fungování Kerbera je funkční překlad doménových jmen na IP adresy. Ten je v běžných sítích zajištěn serverem DNS. Konfigurace serveru DNS je komplexní problematika, více v [19] a [3].

Na jednoduché síti lze použít soubor */etc/hosts*, do kterého lze zadat IP adresu a k ní příslušné doménové jméno počítače v síti. V následujícím postupu bude používán server *server.bp.cz* s IP adresou 192.168.1.1 a klient *klient.bp.cz* s IP adresou 192.168.1.2. Zadáme tedy příslušné hodnoty do souboru */etc/hosts* na obou strojích:

```
192.168.1.1       server.bp.cz      server
192.168.1.2       klient.bp.cz      klient
```

Kerberos pracuje na principu lístků, které přidělují uživatelům oprávnění. Tyto lístky mají v sobě časové razítko, které určuje dobu platnosti tohoto oprávnění. Proto je pro správnou funkčnost důležité synchronizovat čas všech stanic v síti a to nejlépe pomocí NTP serveru [19].

NTP server

Nainstalujeme balíček s NTP serverem:

```
# aptitude install ntp
```

Editujeme soubor */etc/ntp.conf* a zvolíme vhodné servery[10], se kterými se bude náš server časově synchronizovat:

```
server 0.cz.pool.ntp.org
server 1.cz.pool.ntp.org
server 2.cz.pool.ntp.org
server 3.cz.pool.ntp.org
```

Výše uvedenými servery nahradíme servery přednastavené a povolíme počítačům v naší síti synchronizaci s NTP serverem:

```
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

Restartujeme NTP server:

```
# /etc/init.d/ntp restart
```

A příkazem *ntpq* synchronizujeme čas se servery v Internetu [27]:

```
# ntpq –p
```

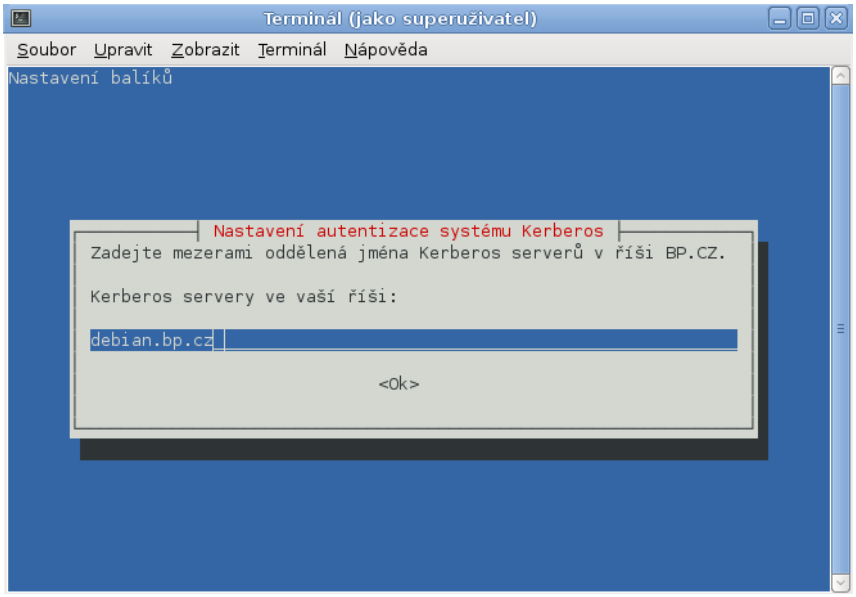
Kerberos server

Nainstalujeme standardním způsobem server Kerberos:

```
# aptitude install rsyslog krb5-{admin-server,user,doc}
```

[10] http://www.pool.ntp.org/zone/cz

Vytvoříme říši s názvem BP.CZ a zvolíme, jaké Kerberos servery jsou v říši přítomny. V naší síti je přítomen pouze jeden Kerberos server, a to *server.bp.cz*:



Obrázek 13 – Nastavení serveru v říši Kerberos

Srdcem konfigurace Kerbera je soubor */etc/krb5.conf*, upravíme ho tedy dle našich představ:

```
[realms]
BP.CZ = {
kdc = server.bp.cz:88
admin_server = server.bp.cz:749
default_domain = bp.cz
}

[domain_realm]
.bp.cz = BP.CZ
bp.cz = BP.CZ

[libdefaults]
default_realm = BP.CZ
dns_lookup_realm = false
dns_lookup_kdc = false

[kdc]
profile = /etc/krb5kdc/kdc.conf

[logging]
default = FILE:/var/log/kerberos/krb5libs.log
kdc = FILE:/var/log/kerberos/krb5kdc.log
admin_server = FILE:/var/log/kerberos/kadmind.log
```

Vytvoříme adresář s logovacími soubory, které jsme specifikovali výše [28]:

```
# mkdir /var/log/kerberos
# touch /var/log/kerberos/krb5libs.log
# touch /var/log/kerberos/krb5kdc.log
# touch /var/log/kerberos/kadmind.log
```

Znemožníme možnost preautentizace[11] tím, že upravíme soubor */etc/krb5kdc/kdc.conf* podle následujícího vzoru:

```
[kdcdefaults]
    kdc_ports = 750,88

[realms]
    BP.CZ = {
        database_name = /var/lib/krb5kdc/principal
        admin_keytab = FILE:/etc/krb5kdc/kadm5.keytab
        acl_file = /etc/krb5kdc/kadm5.acl
        key_stash_file = /etc/krb5kdc/stash
        kdc_ports = 750,88
        max_life = 10h 0m 0s
        max_renewable_life = 7d 0h 0m 0s
        master_key_type = des3-hmac-sha1
        supported_enctypes =aes256-cts:normal
        des3-hmac-sha1:normal des-cbc-crc:normal
    }
```

Vygenerujeme databázi (tento proces může trvat i několik minut). Dále bude nutno zadat klíč, kterým se bude kódovat celá databáze:

```
# krb5_newrealm
```

Povolíme administraci serveru pro všechny principály s instancí admin upravením přístupového seznamu */etc/krb5kdc/kadm5.acl*:

```
*/admin*
admin *
```

Server restartujeme:

```
# /etc/init.d/krb5-admin-server restart
# /etc/init.d/krb5-kdc restart
```

Dále musí být vytvořen principál administrátora:

```
# kadmin.local -q "addprinc krbadmin/admin"
```

Spustíme konfigurační nástroj Kerbera a vytvoříme principál nového uživatele:

```
# kadmin -p krbadmin/admin

kadmin: addprinc novy_uzivatel
```

V případě správného postupu se po zadání příkazu *kinit novy_uzivatel && klist && kdestroy* vypíše [29]:

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: uzivatel@BP.CZ

Valid starting     Expires            Service principal
04/11/12 12:52:03  04/11/12 22:52:03  krbtgt/BP.CZ@BP.CZ
      renew until 04/12/12 12:52:00
```

[11] Preautentizace umožňuje navázat určitou relaci ještě před samotnou autentizací uživatele.

Kerberos klient

Nainstalujeme balíčky pro klienta serveru Kerberos:

```
# aptitude install krb5-user libpam-krb5
```

Upravíme soubor */etc/krb5.conf*, aby odpovídal námi zvolené říši:

```
[realms]
BP.CZ = {
kdc = server.bp.cz:88
admin_server = server.bp.cz:749
default_domain = bp.cz
}

[domain_realm]
.bp.cz = BP.CZ
bp.cz = BP.CZ

[libdefaults]
default_realm = BP.CZ
dns_lookup_realm = false
dns_lookup_kdc = false

[logging]
default = FILE:/var/log/kerberos/krb5libs.log
```

LDAP server

Nainstalujeme balíčky LDAP serveru, při instalaci je nutno zadat heslo pro administrátorský záznam v adresáři LDAP:

```
# aptitude install rsyslog slapd ldap-utils ldapscripts
```

Vytvoříme databázi uživatelů a skupin. Pokud je na serveru k dispozici grafické prostředí, lze si vytváření a správu LDAP databáze usnadnit nástroji s GUI jako jsou JXplorer, Luma, atd. Pokud server disponuje pouze terminálovým přístupem, vytvoříme soubor *init.ldif* a vložíme do něj:

```
dn: dc=bp,dc=cz
objectClass: organization
objectClass: dcObject
objectClass: top
dc: bp
o: Bp

dn: ou=people,dc=bp,dc=cz
objectClass: organizationalUnit
objectClass: top
ou: people

dn: ou=groups,dc=bp,dc=cz
objectClass: organizationalUnit
objectClass: top
ou: groups
```

Soubor aplikujeme [24]:

```
# ldapadd -cxWD cn=admin,dc=bp,dc=cz -f init.ldif
```

Důležité je nastavit i na straně serveru LDAP klienta!

LDAP klient

Nainstalujeme potřebné balíčky:

```
# aptitude install ldap-utils libpam-ldap libnss-ldap
```

Během instalace je potřeba vyplnit údaje o LDAP serveru:

- LDAP server URI: *ldap://server.bp.cz/*
- Rozlišovací název prohledávaného stromu: *dc=bp,dc=cz*
- Verze LDAP: *3*
- LDAP účet uživatele root: *cn=admin,dc=bp,dc=cz*
- LDAP heslo uživatele root:
- Povolit správcovskému LDAP účtu, aby se choval jako lokální root: *Ne*
- Vyžaduje LDAP databáze přihlášení: *Ne*

Editujeme soubor */etc/ldap/ldap.conf*, aby věděl o našem LDAP serveru:

```
BASE        dc=bp,dc=cz
URI         ldap://server.bp.cz
SASL_MECH   GSSAPI
```

Přidáme podporu pro přihlašovaní pomocí LDAP. Editujeme soubor */etc/nsswitch.conf* a přidáme hodnotu *ldap* direktivám *passwd, group* a *shadow*:

```
passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap
```

Nakonec přidáme do souboru */etc/pam.d/common-session* direktivu, která zařídí vytvoření domácího adresáře uživatele při jeho prvním přihlášení [31]:

```
session  required  pam_mkhomedir.so      skel=/etc/skel/   umask=0077
```

Kerberos a Samba

Pro spolupráci Kerbera a Samby musíme Sambě vytvořit, ostatně jako každé kerberizované síťové službě, principál. Spustíme tedy Kerberův administrační nástroj a zadáme [32]:

```
kadmin: addprinc -randkey cifs/server.bp.cz
kadmin: ktadd -k /etc/krb5.keytab -e rc4-hmac:normal cifs/server.bp.cz
```

Upravíme sekci *[global]* souboru */etc/samba/smb.conf* tak, aby Samba používala kerberovské ověřovací mechanismy [33]. Důležité jsou zejména direktivy *realm* a *kerberos method* [32]:

```
[global]
   workgroup = BP
   server string = %h server
   dns proxy = nocast
   interfaces = eth1
   bind interfaces only = yes
   log file = /var/log/samba/log.%m
   max log size = 1000
   syslog = 0
   panic action = /usr/share/samba/panic-action %d

   security = user
   realm = BP.CZ
   kerberos method = system keytab
   encrypt passwords = true
   use spnego = yes

   unix extensions = yes
   case sensitive = yes
   delete readonly = yes
   ea support = yes
```

Nyní má uživatel přístup ke sdíleným adresářům, které jsou poskytovány Sambou.

LDAP server s autentifikací pomocí systému Kerberos

Stejně jako v případě kerberizování Samby, přidáme LDAP principál. V kerberovském administračním nástroji zadáme [34]:

```
kadmin: addprinc -randkey ldap/server.bp.cz
kadmin: ktadd ldap/server.bp.cz
```

Aby měl LDAP server přístup ke klíči, musíme provést úpravu přístupových práv k souboru */etc/krb5.keytab* [35]:

```
# chgrp openldap /etc/krb5.keytab
# chmod g+r,o= /etc/krb5.keytab
```

Jelikož LDAP nepodporuje Kerbera přímo, spoléhá se na SASL. SASL je autentizační framework, který vyděluje autentizační mechanizmus z aplikačního protokolu. Nainstalujeme ho na server i klienta:

```
# aptitude install libsasl2-modules-gssapi-mit libsasl2-2
```

Pro zajištění mapování identit mezi LDAP a SASL vytvoříme soubor *auth-kerberos.ldif* s následujícím obsahem:

```
dn: cn=config
changetype: modify
#
add: olcAuthzRegexp
olcAuthzRegexp: uid=([^,]+),cn=bp.cz,cn=gssapi,cn=auth
```

```
  uid=$1,ou=people,dc=bp,dc=cz
-
#kerberos rise
add: olcSaslRealm
olcSaslRealm: BP.CZ
```

Aplikujeme [35]:

```
# ldapmodify -QY EXTERNAL -H ldapi:/// -f auth-kerberos.ldif
```

Přidávání uživatelů do centrální databáze

Při uvedeném nastavení jsou informace o uživatelích uloženy v LDAP, ale o hesla se stará Kerberos. Nejdříve tedy vytvoříme uživatele v LDAP. Toho docílíme načtením souboru LDIF s definicí uživatele. Vytvoříme tedy soubor *user.ldif* a pomocí něj vložíme do databáze uživatele se jménem *uzivatel*:

```
#skupina
dn: cn=uzivatel,ou=groups,dc=bp,dc=cz
objectClass: posixGroup
objectClass: top
cn: uzivatel
gidNumber: 10000

#ucet
dn: uid=uzivatel,ou=people,dc=bp,dc=cz
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: posixAccount
objectClass: person
objectClass: top
cn: Jmeno Uzivatele
displayName: Jmeno
gidNumber: 20001
givenName: Jmeno
homeDirectory: /home/uzivatel
loginShell: /bin/bash
sn: Uzivatel
uid: uzivatel
uidNumber: 10001
```

Soubor aplikujeme [32]:

```
# ldapadd -f user.ldif
```

Pro import lokálních unixových účtu do LDAP lze použít skripty uvedené v příloze D.

```
# bash importUcty.sh
# bash importSkupiny.sh
```

Výsledný soubor aplikujeme stejně jako v případě manuálního vytvoření uživatele.

Na administračním Kerberos serveru přidáme uživatele:

```
# kadmin.local -q "addprinc uzivatel"
```

Na požádání vložíme heslo, kterým se bude uživatel přihlašovat do systému [33].

4.6.6 SWAT

SWAT je grafické webové rozhraní pro správu Samby, které umožňuje úpravu konfiguračních souborů Samby v příjemném uživatelském prostředí.

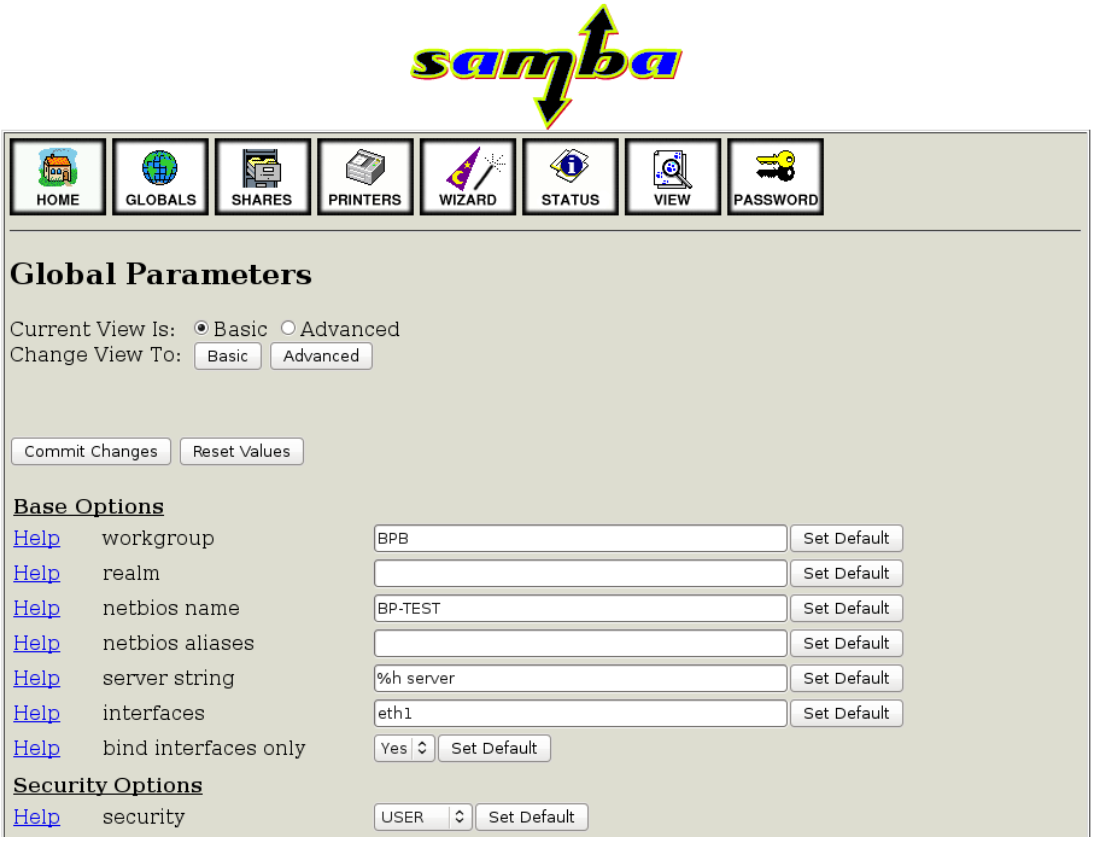
Jeho instalaci provedeme standardním způsobem:

```
# aptitude install swat
```

Adresa pro přístup k rozhraní:

```
http://localhost:901
```

Následně již lze upravovat konfigurační soubory Samby přes připravené formuláře [18].



Obrázek 14 – Rozhraní SWAT

Závěr

Cílem práce byla konfigurace serveru na platformě Debian GNU/Linux, jehož služby byly zajištěny poštovním serverem Postfix a souborovým serverem Samba. Dalším cílem bylo vytvoření centrální autentizace uživatelů, která byla realizována pomocí síťové databáze LDAP a autentizačního serveru Kerberos. V příloze je k dispozici disk virtuálního počítače ve formátu OVA, jehož obsah koresponduje s popisovanými službami.

Pro konfiguraci poskytovaných služeb bylo nutno se seznámit se základní administrací unixových operačních systému v [4]. První kapitola byla věnována obecnému seznámení se serverovými operačními systémy a s požadavky na ně kladené. Druhá kapitola obsahuje informace o operačním systému GNU/Linux se zaměřením na obecný přehled architektury linuxového jádra a způsobu, jakým jádro pracuje s moduly.

První konfigurovanou službou byl poštovní server Postfix, který je velmi flexibilní a může mít při práci s příchozí i odchozí poštou nejrůznější role. V práci byl nakonfigurován jako MTA pro poskytování služeb v rámci lokální sítě pro unixové uživatele serveru. Problematika, jenž popisuje způsob fungování elektronické pošty a způsob jejího přenosu v rámci sítě i v rámci vnitřní architektury Postfixu, byla popsána v teoretické části třetí kapitoly. Praktická část byla věnována konfiguraci samotného Postfixu, ale i IMAP serveru Courier, který umožňuje uživatelům používat běžné klienty MUA pro přístup k poště a poskytuje veškeré výhody protokolu IMAP, jenž byly popsány v teoretické části. Hlavním zdrojem informací o elektronické poště obecně i o serveru Postfix byl [10].

Souborový server Samba, který byl představen ve čtvrté části, umožňuje sdílení souborů a tiskáren i pro klienty s operačními systémy od firmy Microsoft. Díky tomu mohou mít uživatelé své dokumenty k dispozici na všech počítačích, které jsou součástí dané pracovní skupiny či domény. Součástí práce je i podrobný popis konfigurace sdílení domovských adresářů, který má každý uživatel soukromý, obecných sdílených adresářů, které jsou dostupné pro definované skupiny uživatelů, a také sdílení tiskáren. Další možnou konfigurací Samby je využít ji jako server řadiče domény NT. Tato doména je ovšem již značně zastaralá a není možno centrálně ověřovat uživatele unixových systémů. Největším zdrojem informací o souborovém serveru Samba byla [18], o jejíž kvalitě svědčí to, že tuto publikaci převzala komunita vývojářů jako oficiální dokumentační příručku.

Součástí třetí kapitoly je také popis protokolu LDAP a autentizačního serveru Kerberos. Protokol LDAP poskytuje přístup k adresářovým službám. LDAP v konfiguraci, které se práce věnuje, slouží jako centrální autentizační databáze uživatelů a skupin. LDAP tak dokáže zcela plnohodnotně nahradit klasickou databázi uživatelů a skupin unixového typu. V síti mohou existovat také záložní LDAP servery, které zrcadlí databázi hlavního autentizačního serveru, a tak je i v případě jeho výpadku zajištěno korektní přihlašování uživatelů.

Autentizační protokol Kerberos je ve spojení s jinými síťovými službami velice silným nástrojem. Základní popis způsobu jeho fungování byl uveden v kapitole 4.4. Jeho

konfigurace není snadná, ale při správné spolupráci se Sambou, LDAP a dalšími službami vytváří pro uživatele velice efektivní pracovní prostředí. To díky centrálnímu ověřování uživatelů, systému jednotného přihlašování, sdílení tiskáren a domovských i obecných adresářů. Systém jednotného přihlašování, kterým disponuje Kerberos, je velmi silnou a efektivní funkcí. Umožňuje, aby se uživatelé autentizovali v jednotlivých službách, které jsou nakonfigurovány pro spolupráci s Kerberem ihned při přihlášení do samotného systému. Tím tak odpadá nutnost uživatelů přihlašovat se zvlášť ke každé službě.

Všechny představené služby v teoretické části práce byly úspěšně nakonfigurovány a jsou použitelné i pro demonstraci při výuce. Jako jednu z možných modifikací by bylo vhodné šifrovat provoz protokoly TLS a SASL, jak na poštovním, tak na souborovém serveru, jelikož komunikace po síti probíhá v nezašifrované podobě. Pro účely výuky počítačových sítí je ovšem možnost odposlechu komunikace programy jako je Wireshark velice zajímavá. Další vhodnou budoucí modifikací se jeví konfigurace serverů Postfix a Courier pro spolupráci s adresářovou službou LDAP a autentizačním serverem Kerberos.

Při tvorbě práce jsem si výrazně prohloubil znalosti týkající se správy operačního systému GNU/Linux a získal nové zkušenosti s konfigurací serverových služeb.

Literatura

1. MRÁZEK, Luboš. Prednasky predmetu Operacni systemy pro obor Vypocetni technika - ucitelstvi vseobecne vzdelavacich predmetu & Vypocetni technika - bakalarske studium. *Homen.vsb.cz* [online]. [cit. 2012-03-11]. Dostupné z: http://homen.vsb.cz/~kod31/vyuka/opsys/os.html

2. KOLÁŘ, Petr. Operační systémy. *Nti.tul.cz* [online]. [2005] [cit. 2012-03-11]. Dostupné z: http://www.nti.tul.cz/~kolar/os/os-s.pdf

3. KRČMÁŘ, Petr. *Linux*: *postavte si počítačovou síť*. 1. vyd. Praha: Grada, 2008, 182 s. ISBN 978-80-247-1290-1.

4. *Linux*: *dokumentační projekt*. 4., aktualiz. vyd. Překlad Lubomír Ptáček. Brno: Computer Press, 2007, 1334 s. ISBN 978-80-251-1525-1 (VáZ.).

5. KRČMÁŘ, Petr. Historie operačního systému GNU/Linux. *Root.cz* [online]. 2010-06-11 [cit. 2012-03-11]. Dostupné z: http://www.root.cz/texty/historie-operacniho-systemu-gnulinux

6. Definice svobodného software. *Gnu.org* [online]. 2011-12-30 [cit. 2012-03-12]. Dostupné z: http://www.gnu.org/philosophy/free-sw.cs.html

7. GELNER, Radim. Distribuce Linuxu. *Linux.cz* [online]. 2001-04-10 [cit. 2012-03-14]. Dostupné z: http://www.linux.cz/distribuce.html

8. HORÁLEK, Josef. Architektura Linux. *Horalek.org* [online]. [2011] [cit. 2012-03-14]. Dostupné z: http://horalek.org/fim/0S/09.pdf

9. HORÁLEK, Josef. Architektura jádra. *Horalek.org* [online]. [2011] [cit. 2012-03-14]. Dostupné z: http://horalek.org/fim/OS2/Lecture12.pdf

10. DENT, Kyle D. *Postfix*: *kompletní průvodce*. 1. vyd. Praha: Grada, 2005, 237 s. ISBN 80-247-1029-3.

11. JELÍNEK, Lukáš. Stavíme poštovní server – 1 (Postfix). *Abclinuxu.cz* [online]. 2009-10-12 [cit. 2012-03-14]. Dostupné z: http://www.abclinuxu.cz/clanky/site/stavime-postovni-server-1-postfix

12. PETERKA, Jiří. Elektronická pošta v Internetu. *Earchiv.cz* [online]. [1998] [cit. 2012-03-14]. Dostupné z: http://www.earchiv.cz/a98/a805t200.php3

13. Postfix. *Wiki.debian.org* [online]. 2011-08-03 [cit. 2012-03-16]. Dostupné z: http://wiki.debian.org/Postfix

14. JELÍNEK, Lukáš. Stavíme poštovní server – 3 (instalace, základní konfigurace Postfixu). *Abclinuxu.cz* [online]. 2009-10-30 [cit. 2012-03-16]. Dostupné z:

http://www.abclinuxu.cz/clanky/site/stavime-postovni-server-3-instalace-zakladni-konfigurace-postfixu

15. Postfix HOWTO. *Wiki.centos.org* [online]. 2011-06-15 [cit. 2012-03-17]. Dostupné z: http://wiki.centos.org/HowTos/postfix

16. Postfix Basic Configuration. *Postfix.org* [online]. [cit. 2012-03-17]. Dostupné z: http://www.postfix.org/BASIC_CONFIGURATION_README.html

17. Courier. *Help.ubuntu.com* [online]. 2011-05-03 [cit. 2012-03-17]. Dostupné z: https://help.ubuntu.com/community/Courier

18. ECKSTEIN, Robert. *Samba*: *Linux jako server v sítích Windows*. Vyd. 2. Brno: Computer Press, 2005, 525 s. ISBN 80-251-0649-7.

19. SMITH, Roderick W. *Linux ve světě Windows*: *průvodce administrátora heterogenních sítí*. 1. vyd. Praha: Grada, 2006, 443 s. ISBN 80-247-1470-1.

20. Samba and the PFIF. *Samba.org* [online]. 2007-12-20 [cit. 2012-03-22]. Dostupné z: http://www.samba.org/samba/PFIF

21. JAKUBČÍK, Ondřej. Jak se tančí Samba?. *Linuxexpress.cz* [online]. 2007-05-16 [cit. 2012-03-22]. Dostupné z: http://www.linuxexpres.cz/praxe/jak-se-tanci-samba/

22. Jaký je rozdíl mezi doménou, pracovní skupinou a domácí skupinou?. *Windows.microsoft.com* [online]. [2012] [cit. 2012-03-23]. Dostupné z: http://windows.microsoft.com/cs-CZ/windows7/What-is-the-difference-between-a-domain-a-workgroup-and-a-homegroup

23. Síťové protokoly (XVI. část), Protokol SMB. *Owebu.bloger.cz* [online]. 2008-02-16 [cit. 2012-03-23]. Dostupné z: http://owebu.bloger.cz/PC-site/Sitove-protokoly-XVI-cast-Protokol-SMB?km=b

24. MLÍKA, Jiří. Kerberos a SSO: Jednotné účty v LDAP. *Abclinuxu.cz* [online]. 2008-05-20 [cit. 2012-04-11]. Dostupné z: http://www.abclinuxu.cz/clanky/bezpecnost/kerberos-a-sso-jednotne-ucty-v-ldap

25. HONTAÑÓN, Ramón J. *Linux*: *praktická bezpečnost.* 1. vyd. Praha: Grada, 2003, 438 s. ISBN 80-247-0652-0.

26. NetworkPrintingWithUbuntu. *Help.ubuntu.com* [online]. 2012-03-24 [cit. 2012-03-23]. Dostupné z: https://help.ubuntu.com/community/NetworkPrintingWithUbuntu

27. NTP Server. *Server-world.info* [online]. 2011-03-05 [cit. 2012-04-10]. Dostupné z: http://www.server-world.info/en/note?os=Debian_6.0&p=ntp

28. KORNATSKYY, Andriy. Debian Kerberos Master. *Mindref.blogspot.com* [online]. 2010-12-21 [cit. 2012-04-12]. Dostupné z: http://mindref.blogspot.com/2010/12/debian-kerberos-master.html

29. MLÍKA, Jiří. Kerberos: přihlašování snadno a rychle. *Abclinuxu.cz* [online]. 2008-05-09 [cit. 2012-04-09]. Dostupné z: http://www.abclinuxu.cz/clanky/bezpecnost/kerberos-prihlasovani-snadno-a-rychle

30. KORNATSKYY, Andriy. Debian OpenLDAP client with Kerberos. *Mindref.blogspot.com* [online]. 2011-02-05 [cit. 2012-04-12]. Dostupné z: http://mindref.blogspot.com/2011/02/debian-openldap-kerberos-client.html

31. SambaKerberos. *Help.ubuntu.com* [online]. 2011-07-04 [cit. 2012-04-13]. Dostupné z: https://help.ubuntu.com/community/Samba/Kerberos

32. MLÍKA, Jiří. Kerberos a SSO: Samba. *Abclinuxu.cz* [online]. 2008-07-24 [cit. 2012-04-10]. Dostupné z: http://www.abclinuxu.cz/clanky/bezpecnost/kerberos-a-sso-samba

33. MLÍKA, Jiří. Kerberos a LDAP. *Abclinuxu.cz* [online]. 2008-09-24 [cit. 2012-04-13]. Dostupné z: http://www.abclinuxu.cz/clanky/bezpecnost/kerberos-a-ldap

34. KORNATSKYY, Andriy. Debian OpenLDAP with Kerberos Authentication. *Mindref.blogspot.com* [online]. 2011-02-05 [cit. 2012-04-14]. Dostupné z: http://mindref.blogspot.com/2011/02/debian-openldap-kerberos-authentication.html

Příloha A – Testování funkčnosti protokolu SMTP a IMAP

Funkčnost protokolu SMTP a tím i Postfixu můžeme otestovat vytvořením a zasláním zprávy přes Telnet.

Připojíme se k místnímu počítači přes port 25, který slouží právě Telnetu:

```
telnet localhost 25
```

A vytvoříme novou zprávu:

```
mail from:<nekdo@domena.cz>
rcpt to:<sretr@bp.cz>
data
To: sretr@bpbp.cz
From: nekdo@domenabp.cz
Subject: test smtp
Dorazil email ?
.
quit
```

Zde vidíme v praxi, že sekce To: a From: jsou opravdu součástí těla zprávy, a jejich hodnota nemá na adresu odesílatele, resp. příjemce, žádný vliv. Tečka na samostatném řádku je důležitá, pomocí ní ukončíme zadávání zprávy.

Poté stačí v příkazovém interpretu zadat příkaz *mail* a zkontrolovat, zda zpráva dorazila [13].

Testování protokolu IMAP probíhá obdobně:

```
telnet localhost 143
```

Vypíše se:

```
Trying localhost...
Connected to localhost.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE
THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT QUOTA IDLE ACL ACL2=UNION]
Courier-IMAP ready. Copyright 1998-2010 Double Precision, Inc.  See
COPYING for distribution information.
```

Přihlásíme se pod našim uživatelským jménem a heslem:

```
imap login uzivatelske_jmeno heslo
```

V případě, že přihlášení proběhlo úspěšně, vypíše se [17]:

```
imap OK LOGIN Ok.
```

Příloha B – Vzorový konfigurační soubor main.cf

```
#konfiguracni soubor pro Postfix v ramci LAN
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)

myhostname = server.bp.cz

#myorigin stanovi domenovy nazev, pokud neni uveden v obalce
myorigin = bp.cz

mydomain = $myorigin

#mydestination stanovuje domenove nazvy lokalniho doruceni
mydestination = $myhostname localhost.$mydomain localhost $mydomain

#my_networks_style = subnet znamena, ze postfix bude preposilat postu
vsem smtp klientum na stejne podsiti jako se nachazi server
mynetworks_style = subnet

#realyhost urcuje, ktere servery se budou starat o zpravy mimo lokalni
domenu
relayhost =

#realy_domains oznacuje domeny pro ktere bude postfix preposilat zpravy
relay_domains =

inet_interfaces = all

inet_protocols = ipv4

mailbox_command =

mailbox_size_limit = 0

recipient_delimiter = +

default_transport = error

relay_transport = error

alias_maps = hash:/etc/aliases

alias_database = hash:/etc/aliases

#home_mailbox urcuje adresar do ktereho se bude lokalnim uzivatelum
ukladat posta
home_mailbox = Maildir/
```

Příloha C – Vzorový konfigurační soubor smb.conf

```
#konfiguracni soubor pro sambu, ktera nabizi sdilene adresare, tiskarny a
#centralni databazi uzivatelu pres LDAP a Kerberos

 [global]
      workgroup = BP
      server string = %h server
      dns proxy = nocast
      interfaces = eth1
      bind interfaces only = yes
      log file = /var/log/samba/log.%m
      max log size = 1000
      syslog = 0
      panic action = /usr/share/samba/panic-action %d
      #security urcuje zpusob rizeni pristupu, viz kapitola 4.2.1
      security = user
      #realm je nutny pro fungovani kerbera, urcuje jeho risi
      realm = BP.CZ
      #kerberos method je nutne pro ziskani listku od kerbera
      kerberos method = system keytab
      encrypt passwords = true
      #spnego je nutne pouzit pokud se klient autentizuje na serveru, ale
      predem neni jasno jakym protokolem
      use spnego = yes
      case sensitive = yes
      delete readonly = yes
      ea support = yes
      load printers = Yes
      #maska nastavuje maximalni pripustna prava novych souboru a
      adresaru, pouzit je unixovy zapis prav, vice o nem v [4]
      create mask = 0640
      directory mask = 0750
      nt acl support = No
      printing = cups
      printcap name = cups


[homes]
      comment = Home Folder
      valid users = %S
      read only = No
      create mask = 0750
      #browseable urcuje zda je svazek viditelny pri prohlizeni site
      browseable = No
```

```
      #volume specifikuje navesti svazku
      volume = %U Home

[printers]
      comment = Network Printers
      guest ok = no
      printable = yes
      path = /var/spool/samba
      browseable = No
      read only  = Yes
      printable = Yes
      #print command = /usr/bin/lpr -P%p -r %s

[print$]
       path = /home/printers
       guest ok = No
       browseable = Yes
       read only = Yes
      #valid users urcuje, kteri uzivatele maji pristup k danemu svazku
       valid users = @"Print Operators"
       write list = @"Print Operators"
       create mask = 0664
       directory mask = 0775

[sdileny_adresar]
      comment = Sdilena data uzivatelu
      path = /var/lib/samba/sdileny_adresar
      read only = No
      create mask = 0660
      directory mask = 0770
      public = yes
```

Příloha D – Import unixových účtů a skupin do LDAP

```
#!/bin/bash
#Import unixovych uctu
SUFFIX='dc=bp,dc=cz'
LDIF='ldapuser.ldif'

echo -n > $LDIF
for line in `grep "x:[1-9][0-9][0-9][0-9]:" /etc/passwd | sed -e "s/
/%/g"`
do
    UID1=`echo $line | cut -d: -f1`
    NAME=`echo $line | cut -d: -f5 | cut -d, -f1`
    if [ ! "$NAME" ]
    then
        NAME=$UID1
    else
        NAME=`echo $NAME | sed -e "s/%/ /g"`
    fi
    SN=`echo $NAME | awk '{print $2}'`
    if [ ! "$SN" ]
    then
        SN=$NAME
    fi
    GIVEN=`echo $NAME | awk '{print $1}'`
    UID2=`echo $line | cut -d: -f3`
    GID=`echo $line | cut -d: -f4`
    PASS=`grep $UID1: /etc/shadow | cut -d: -f2`
    SHELL=`echo $line | cut -d: -f7`
    HOME=`echo $line | cut -d: -f6`
    EXPIRE=`passwd -S $UID1 | awk '{print $7}'`
    FLAG=`grep $UID1: /etc/shadow | cut -d: -f9`
    if [ ! "$FLAG" ]
    then
        FLAG="0"
    fi
    WARN=`passwd -S $UID1 | awk '{print $6}'`
    MIN=`passwd -S $UID1 | awk '{print $4}'`
    MAX=`passwd -S $UID1 | awk '{print $5}'`
    LAST=`grep $UID1: /etc/shadow | cut -d: -f3`

    echo "dn: uid=$UID1,ou=people,$SUFFIX" >> $LDIF
    echo "objectClass: inetOrgPerson" >> $LDIF
    echo "objectClass: organizationalPerson" >> $LDIF
    echo "objectClass: posixAccount" >> $LDIF
    echo "objectClass: person" >> $LDIF
    echo "objectClass: top" >> $LDIF
    echo "cn: $NAME" >> $LDIF
    echo "displayName: $NAME" >> $LDIF
    echo "gidNumber: $GID" >> $LDIF
    echo "givenName: $GIVEN" >> $LDIF
    echo "homeDirectory: $HOME" >> $LDIF
    echo "loginShell: $SHELL" >> $LDIF
    echo "sn: $SN" >> $LDIF
    echo "uid: $UID1" >> $LDIF
    echo "uidNumber: $UID2" >> $LDIF
    echo >> $LDIF
done
```

```
#!/bin/bash
#Import unixovych skupin
SUFFIX='dc=bp,dc=cz'
LDIF='ldapgroup.ldif'

echo -n > $LDIF
for line in `grep "x:[1-9][0-9][0-9][0-9]:" /etc/group`
do
    CN=`echo $line | cut -d: -f1`
    GID=`echo $line | cut -d: -f3`
    echo "dn: cn=$CN,ou=groups,$SUFFIX" >> $LDIF
    echo "objectClass: posixGroup" >> $LDIF
    echo "objectClass: top" >> $LDIF
    echo "cn: $CN" >> $LDIF
    echo "gidNumber: $GID" >> $LDIF
    users=`echo $line | cut -d: -f4 | sed "s/,/ /g"`
    for user in ${users} ; do
        echo "memberUid: ${user}" >> $LDIF
    done
    echo >> $LDIF
done
```

Příloha E – DVD s obrazem disku virtuálního počítače

Přiložený disk DVD obsahuje obraz disku virtuálního počítače s popisovanou konfigurací ve formátu OVA.