Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Bakalářská práce

Administrace SUSE Linuxu

Jiří Malák

! ! !

Místo této strany vložíte zadání bakalářské práce.
(Do jedné vazby originál a do druhé kopii)

! ! !

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Administrace SUSE Linuxu" jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30.4.2009 _____

Poděkování

Rád bych touto cestou poděkoval panu doc. Ing. Arnoštovi Veselému CSc. za pozornost, kterou věnoval mé práci a za jeho odborné rady při vypracování této bakalářské práce.

Administrace SUSE Linuxu

SUSE Linux administration

Souhrn

Tato práce se zabývá popisem administrace operačního systému Linux. Konkrétně se zaměřuje na distribuce SUSE Linux, které vyvíjí firma Novell. Práce je určena zejména pro začínající správce se základní znalostí OS Linux. Zaměřuje se hlavně na základní správu systému a nastavení některých důležitých serverových služeb (jako WWW, DHCP, DNS a Samba). Práce je rozdělena do pěti částí.

První část pojednává o správě uživatelů. V této části je popsán mechanismus správy uživatelů a skupin v operačním systému Linux. Následující část je věnována správě souborového systému, která se zabývá správou pevných disků, typy souborových systémů, přístupovými právy a bezpečností dat v souborových systémech. Další část popisuje správu software. Tato část se zabývá balíčkovacím systémem, možnostmi instalace software a aktualizací nainstalovaných aplikací. Ve čtvrté části je popsána konfigurace TCP/IP komunikace a některé základní serverové služby. V páté, praktické části je popsána konfigurace domácího serveru. Jsou zde uvedeny hlavní konfigurační soubory služeb, které tento server poskytuje a návod k jejich nastavení.

Summary

This thesis deals with the characterization of the operating system Linux. Specifically is aimed at distribution of SUSE Linux, which is developed by a company Novell. The information is particularly intended for beginning administrators with basic OS Linux knowledge. The project is focused on the basic system administration and the important server services setting (such as WWW, DHCP, DNS and Samba). The work is split up into five parts.

The first one deals with user administration.

In this part, administration of users and groups is described. Following part is given to files system administration, which deals with hard disk administration, types of files system, access rights and a security of system data. Next part describes software administration. This part deals with package system, possibilities of software installation and installed application update. In the fourth part, there is a description of TCP/IP communication configuration and a description of base server services. In the fifth, practical part, there is characterized a configuration of home server. There are mentioned main configuration files for services, which this server offers and guide for their setting.

Klíčová slova: Linux, SUSE, správa, konfigurace, instalace, příkazy, služby, server

Keywords: Linux, SUSE, administration, configuration, installation, commands, services, server

1. Úvod

Téma bakalářské práce „Administrace SUSE Linuxu“ autor zvolil z důvodu zájmu o tento operační systém (dále OS) a možnosti praktického uplatnění nabytých znalostí v budoucnosti. S OS Linux přišel autor poprvé do styku při výuce předmětu Operační systémy II a velice ho zaujal svojí architekturou, stabilitou a možností vlastní modifikace.

Za vytvořením jádra operačního systému Linux stál Linus Torvald, který oslovil širokou programátorskou veřejnost, aby se podílela na vývoji tohoto jádra. Toto jádro dostalo název Linux a bylo spojeno s projektem GNU, který poskytnul základní balík volně šiřitelných programů. Tímto spojením vznikl OS GNU/Linux (dále Linux).

V současné době už není OS Linux pouhou „hračkou“ několika nadšenců, ale je plnohodnotným operačním systémem. Má přívětivé grafické rozhraní a existuje velké množství dostupných softwarových aplikací. Jeho hlavní využití je především v serverech, a to díky vysoké stabilitě, bezpečnosti a rozšířeným možnostem nastavení. To však s sebou přináší větší nároky na znalosti administrátora.

OS Linux je šířen v rámci licenční politiky GNU GPL (General Public Licence), z níž vyplývá, že je dovolena jakákoli změna ve zdrojových kódech, ale změněný software nesmí být uzavřen veřejnosti a distribuován za peníze. To ovšem nevylučuje distribuci otevřeného software za peníze. Díky zmíněné licenční politice proto vzniklo velké množství placených i neplacených distribucí OS Linux, což znesnadňuje orientaci v jednotlivých distribucích a vytváří tím jeden z hlavních problémů OS Linux.

Z tohoto důvodu se tato práce zaměřuje na distribuce SUSE Linux od firmy Novell. Firma Novell poskytuje dvě placené distribuce SUSE Linuxu a to SUSE Linux enterprise desktop a SUSE Linux enterprise server. Obě distribuce jsou vhodné hlavně pro firemní využití. Dále sponzoruje vývoj otevřené distribuce openSUSE. Vývoj openSUSE je realizován celosvětovou komunitou vývojářů, uživatelů a opensource nadšenců, kteří se snaží z distribuce openSUSE udělat jednu

z nejrozšířenějších a nejlepších distribucí na světě. V této práci se budeme zabývat administrací této distribuce.

2. Cíl práce a metodika

2.1. Cíl práce

Cílem této práce je popsat nastavení a správu operačního systému SUSE Linux. V rámci tohoto cíle chce autor nastínit základní správu systému včetně popisu a nastavení základních serverových služeb. Dále si autor klade za cíl implementaci těchto poznatků v podobě příkladu nastavení jeho vlastního domácího serveru.

2.2. Metodika

K vypracování bakalářské práce byly použity dostupné literární a internetové zdroje uvedené v seznamu literatury. Jako hlavní zdroj autor použil internetové zdroje a knihu E.Nemetha, G. Snydera a T. Heina „Kompletní příručka administrátora“, 2008. Čerpáno bylo i ze starších zdrojů protože i přes rychlý vývoj OS Linux, zůstává základní správa téměř nezměněna.

V teoretické části se autor zaměřuje na popis nástrojů a služeb, které jsou dostupné ve většině distribucí OS Linux. Z důvodů rozsáhlosti tématu se autor zaměřil na popis základní správy a na popis základních serverových služeb.

Většina příkazů a nastavení byla prakticky ověřena autorem na operačním systému openSUSE 10.3.

V praktické části autor uvádí vlastní nastavení domácího serveru. Pro ověření nastavení byl použit autorův domácí server s následující hardwarovou konfigurací:

- základní deska: Intel D201GLY2A uATX
- procesor: Intel Celeron 220 1,2GHz
- paměť: 1 GB DDR2 400MHz
- grafická karta: integrovaná
- pevný disk: 160 GB SATA II

Nastavení serveru bylo provedeno z prostředí příkazového řádku.

3. Správa uživatelů

Operační systém Linux používá dva typy identifikace, uživatele a skupiny. Každému názvu uživatele nebo skupiny je přiděleno číslo, které operační systém interně používá místo názvů.

Každý uživatel má v systému přiděleno jednoznačné uživatelské jméno a identifikační číslo. Toto číslo se označuje jako UID.

Použití UID místo uživatelského jména, umožňuje efektivní identifikaci jádrem operačního systému.

Druhým typem identifikace jsou skupiny. Skupiny mají svoje jména a identifikační čísla podobně jako uživatelé. Identifikační číslo skupiny se označuje GID. Každý uživatel musí být členem alespoň jedné skupiny. Jedna z uživatelových skupin je nastavena jako výchozí. Tato skupina se používá například pro vytváření nových souborů a adresářů.

3.1. Uživatelé

V Linuxu obecně existují dvě skupiny uživatelů. První je speciální uživatel - superuživatel (root) a druhá skupina jsou ostatní uživatelé.

3.1.1. Superuživatel

Účet superuživatele je účet pro správu operačního systému. Dovoluje provádět nastavení a operace, které běžným uživatelům nejsou dostupné. Superuživatel má UID = 0. Může provádět jakékoliv přípustné operace v operačním systému bez omezení. Například má přístup ke všem souborům a procesům bez ohledu na aktuálně nastavená přístupová práva. Dále je superuživatel vlastníkem všech systémových souborů a procesů.

Přihlašování do systému přímo na účet root není vhodné. Důvodem je, že se nikde nevytváří záznam o operacích, které byly provedeny v roli superuživatele. Pokud má k superuživatelskému účtu přístup několik uživatelů, není možné zjistit, kdo a kdy ho použil. Proto je dobré zablokovat přihlášení superuživatele na většině terminálů a dovolit přihlášení jen z vybraných terminálů a obecně omezit jeho používání jen na nezbytné aktivity. Většinou se pracuje pod běžným uživatelským

účtem. V případě potřeby může uživatel použít superuživatelský účet pomocí příkazu **su**, který po zadání hesla k superuživatelskému účtu spustí shell superuživatele. Důležité je nezapomenout ukončit tento shell po ukončení činnosti vyžadující superuživatelský účet. Příkaz **su** sice nezaznamenává příkazy, které byly provedeny, ale vytváří záznam, kdo a kdy se stal superuživatelem.

Další možností jak spouštět příkazy se superuživatelskými právy je program **sudo**. **Sudo** umožňuje spouštět jednotlivé příkazy se superuživatelskými právy. Program **sudo** prohlédne soubor `/etc/sudousers`, který obsahuje seznam uživatelů a příkazů, které smějí používat jednotliví uživatelé. Jestliže uživatel má oprávnění příkaz provést, musí zadat svoje heslo a následně bude příkaz proveden. **Sudo** zaznamenává všechny prováděné příkazy včetně uživatele, který tyto příkazy použil. Zaznamenává také adresáře, ze kterých se příkazy spouštěly, a čas, kdy byly provedeny. [1][2]

3.1.2. Ostatní uživatelé

Ostatní uživatelé mohou provádět pouze činnosti, na které mají oprávnění, a pracovat se soubory, ke kterým mají oprávnění. Oprávnění se vyhodnocují operačním systémem na základě UID a GID.

3.2. Základní konfigurační soubory

3.2.1. Soubor /etc/passwd

Soubor `/etc/passwd` (příloha A) obsahuje kompletní seznam uživatelských účtů. Systém při přihlašování prohledá tento soubor, aby ověřil, zda je použité uživatelské jméno a heslo platné. Následně získá UID přihlašovaného uživatele.

Každý řádek v souboru `passwd` představuje jednoho uživatele a obsahuje sedm údajů oddělených dvojtečkami. Tyto údaje jsou přihlašovací jméno, heslo, UID, implicitní GID, informace o uživateli, cestu k domácímu adresáři a přihlašovací shell.

Přihlašovací jméno (neboli uživatelské jméno) musí být jednoznačné. Může obsahovat jakékoliv znaky kromě dvojteček a znaku nový řádek. Rozlišují se velká a malá písmena a délka přihlašovacího jména nesmí být delší než 32 znaků.

V poli heslo může být buď "x", což znamená, že heslo je zašifrované v souboru `/etc/passwd`, nebo "*", což znamená, že heslo není platné.

Třetím údajem je UID. Obecné doporučení je přiřazovat běžným uživatelům UID od 500. UID od 1 – 500 se obvykle používají pro různé systémové účely nebo pro uživatele, kteří nepředstavují fyzické osoby.

GID v souboru passwd je GID implicitní skupiny.

Pole informace o uživateli neboli GECOS field (General Electric Comprehensive Operating Supervisit) se používá pro záznam osobních informací o uživatelích a nemá žádnou obecně definovanou strukturu. Většinou se v tomto poli uvádí celé jméno a příjmení, případně pracovní telefonní číslo či označení kanceláře nebo budovy. Uživatelé si toto pole mohou měnit příkazem **chfn** z důvodu aktualizování informací.

Přihlašovací shell je obvykle příkazový interpret (jako například C shell nebo bash), který je spuštěn po přihlášení do systému. Pokud je toto pole prázdné, spouští se implicitně nastavený shell **/bin/bash**. Uživatelé si mohou svůj shell změnit příkazem **chsh**. Tento shell, ale musí být uveden v souboru `/etc/shells`. Shell, který není uveden v tomto souboru, není možno spustit. [1][3][4]

3.2.2. Soubor /etc/shadow

Soubor `/etc/shadow` (příloha B) slouží k ukrytí zašifrovaných hesel před běžnými uživateli a poskytuje další informace o uživatelských účtech, které nejsou v souboru `passwd`. Z důvodu bezpečnosti hesel tento soubor může měnit a číst pouze superuživatel. Soubor `shadow` má také řádkovou strukturu jako `/etc/passwd`. Každý řádek obsahuje 9 údajů, které jsou odděleny dvojtečkami. Obsahuje přihlašovací jméno, zašifrované heslo, datum poslední změny hesla, minimální počet dnů mezi změnami hesla, maximální počet dnů mezi změnami hesla, počet dnů do vypršení hesla, počet dnů po vypršení hesla, po kterých bude účet zablokován, datum vypršení účtu a jedno pole rezervované pro budoucí použití.

Přihlašovací jméno musí být stejné jako v souboru `passwd`, protože propojuje položky v souborech passwd a shadow.

Uložené heslo je zašifrované. Heslo může být zašifrováno jednou z následujících metod: DES, MD5 a Blowfish.

Datum poslední změny hesla udává počet dní od poslední změny hesla. Toto pole většinou automaticky vyplňuje program **passwd**, který slouží ke změně hesla.

Pole minimální počet dnů mezi změnami hesla udává počet dní, během kterých nemůže uživatel změnit svoje heslo. Z hlediska bezpečnosti to není optimální, a proto se doporučuje nastavit tuto hodnotu na 0.

Pole maximální povolený počet dnů mezi změnami hesla umožňuje správci vynutit změnu hesel.

Sedmé pole udává, kolik dní od vypršení hesla se má čekat, než bude přihlašovací jméno považováno za skutečně vypršené.

Osmé pole udává datum, po jehož uplynutí dojde k vypršení účtu. Účet bude zablokován a uživatel se nebude moci přihlásit, dokud správce pole znovu nenastaví. Když ponecháme toto pole prázdné, účet nikdy nevyprší.

Deváté pole se v současnosti nevyplňuje a je vyhrazeno pro budoucí použití.

Pro sjednocení obsahu souboru `shadow` se souborem `passwd` používáme program **pwconv**. **Pwconv** vyplňuje většinu parametrů souboru shadow z výchozích hodnot, které jsou specifikovány v souboru `/etc/login.defs.` [1][3][4]

3.2.3. Soubor /etc/group

Soubor `/etc/group` obsahuje informace o skupinách v systému. Každý řádek v souboru `group` představuje jednu skupinu a každý řádek má čtyři pole, která jsou oddělena dvojtečkami. Tato pole jsou název skupiny, heslo, číslo skupiny GID a seznam uživatelů, kteří jsou ve skupině (uživatele oddělujeme čárkami). [1]

3.3. Přidání uživatele

Přidat nového uživatele lze dvěma způsoby. Můžeme použít příkazy pro přidání uživatele nebo provést přímé přidání uživatele do příslušných souborů.

3.3.1. Ruční přidání uživatele

Přímé přidání nového uživatele do souborů se většinou nepoužívá z bezpečnostních důvodů. Při editaci souborů může dojít k poškození informací v těchto souborech díky překlepům nebo omylem. Přidání uživatele se skládá z několika kroků, které musí být provedeny superuživatelem. [1]

Upravení souborů passwd a shadow

Pro bezpečnou úpravu souboru `passwd` a `shadow` použijeme program **vipw**, který zajistí, že soubor `passwd` bude upravován jen jedním uživatelem. Soubor `passwd` editujeme příkazem **vipw** bez volby a soubor `shadow` editujeme pomocí volby **–s**. [1]

Přidání uživatele do souboru /etc/group

Pomocí programu **vipw** s volbou **–g** můžeme bezpečně změnit soubor `/etc/group`. Tímto programem přidáme uživatelské jméno do všech skupin, kterých je členem.

Nastavení úvodního hesla

Z důvodu bezpečnosti by neměl být žádný uživatelský účet ponechán bez hesla. Vždy by se mělo nastavit úvodní heslo. Nastavení hesla provedeme příkazem **passwd**. [1]

Vytvoření domácího adresáře uživatele

Po vytvoření nového uživatele je nutné vytvořit jeho domácí adresář. Musí se změnit vlastník adresáře, neboť adresář je vytvořen jiným uživatelem (obvykle superuživatel). Ze stejného důvodu je nutné upravit i přístupová práva k tomuto adresáři. [1]

Startovacích soubory

Startovací soubory začínají tečkou a obvykle končí písmeny rc. Některé nástroje a příkazy můžeme přizpůsobit pomocí startovacích souborů, které jsou umístěny v domácím adresáři uživatele. Je vhodné mít prototypy startovacích souborů uložené na zvláštním místě a jen je překopírovat do domácího adresáře

uživatele a případně je upravit podle potřeby. Dále musíme nastavit vlastníka a přístupová práva pro tyto soubory. [1]

Kontrola nastavení účtu

Správné nastavení nového účtu prověříme nejlépe tak, že se přihlásíme na nový účet. Dále provedeme kontrolu, zda jsme správně nastavili vlastníka a skupiny startovacích souborů. [1]

3.3.2. Přidání uživatele pomocí příkazů

Vytvoření nového účtu a jeho úpravy provádíme pomocí příkazů **useradd** a **usermod**. Tyto příkazy je nutné spouštět jako superuživatel, protože modifikují soubory `/etc/passwd`, `/etc/shadow` a `/etc/group`. Tyto příkazy značně ulehčují postup přidání nového uživatele. Přesto si většinou administrátoři vytvářejí svoje vlastní příkazy, které lépe vyhovují jimi používaným pravidlům a zvyklostem.

Useradd

Vytvoření nového uživatele provádí příkaz **useradd**, který zapíše nového uživatele do souboru `/etc/passwd`, `/etc/shadow` a `/etc/group`. Pokud nespecifikujeme argumenty příkazu **useradd**, je vytvořen účet se standardním nastavením. UID je o jedničku větší než UID posledního přidaného uživatele. Pro ostatní parametry se použije výchozí nastavení, které je uloženo v souboru `/etc/default/useradd`. Příkaz **useradd** může být spuštěn s dalšími volbami. To umožňuje například automaticky vytvořit domácí adresář pro uživatele, definovat nestandardní umístění domácího adresáře, definovat skupiny, do kterých má být uživatel zařazen, nebo jaký interpret příkazů bude mít uživatel nastavený jako výchozí. [1][2][5]

Usermod

Modifikaci již existujícího účtu můžeme provést pomocí příkazu **usermod**. Příkazem **usermod** můžeme například změnit pole GECOS, změnit domácí adresář, nastavit, kdy má skončit platnost účtu, přestěhovat domácí adresář do jiného adresáře a nebo zamknout či odemknout heslo. [2][5]

3.4. Odebrání uživatele

Odebrání uživatele představuje odstranění všech odkazů na uživatelské jméno. Uživatele můžeme odstranit ručně nebo pomocí příkazu **userdel**.

Ruční odstranění uživatele spočívá v odebrání uživatele ze všech lokálních databází a souboru `aliases`, odstranění souboru `crontab` a přechodných souborů v adresářích `/var/tmp` nebo `/tmp`, ukončení všech procesů uživatele, odstranění uživatele ze souborů `/etc/passwd`, `/etc/shadow` a `/etc/group` a odstranění domácího adresáře uživatele. Po odstranění uživatele je vhodné zkontrolovat, jestli staré UID uživatele nevlastní v souborovém systému některé soubory.

Příkazem **userdel** smažeme odkazy na uživatele ze souborů `/etc/passwd`. `/etc/shadow` a `/etc/group`. Příkaz **userdel** ve výchozím nastavení nemaže domácí adresář uživatele, proto pokud nechceme domácí adresář uschovat, je vhodné použít příkaz **userdel** s volbou **–r**. Příkaz **userdel** je sice velmi praktický, ale má jisté nevýhody. Například smaže pouze záznamy v souborech, ale neprovede ostatní důležité kroky, jako třeba ukončení všech procesů uživatele. Neodstraní uživatele ze souboru `aliases` a neodstraní jeho dočasné soubory. [1][5]

3.5. Skupiny uživatelů

Skupiny se využívají na mnoha místech podobně jako UID. Například pro přístup k souborovému systému, přístup k hardware, kontrole síťových služeb, filtrování paketů, routování apod. [2]

3.5.1. Přidání skupiny

Skupinu můžeme přidat vytvořením řádku v souboru `/etc/group` nebo pomocí příkazu **groupadd**. Příkaz **groupadd** automaticky vytvoří řádek v souboru `/etc/group` a přiřadí GID o jedničku větší než je GID poslední přidané skupiny. Po vytvoření skupiny přidáme pomocí příkazu **usermod** vybrané uživatele do dané skupiny. [4][5]

3.5.2. Odebrání skupiny

Odstranit skupinu můžeme buď smazáním řádku v souboru `/etc/group` nebo použitím příkazu **groupdel**. Při mazání skupiny je nutno si uvědomit, že smazáním skupiny nesmažeme uživatelské účty, které jsou ve skupině, ani soubory, které byly touto skupinou vlastněny. Proto je nutné provést ruční kontrolu a případné pozůstatky odstranit. [4][5]

3.6. Bezpečnost hesel

Nejčastější bezpečnostní slabinou je nedostatečná správa hesel. Soubory `/etc/passwd` a `/etc/shadow` určují, kdo se smí přihlásit do systému, a proto je důležité, aby byly správně udržovány a kontrolovány.

To, že si každý uživatel může zvolit vlastní heslo, vede k mnoha problémům s bezpečností. Proto je vhodné při přidání nového uživatelského účtu uživateli poskytnout rady pro správný výběr hesla. Nevhodná jsou hesla, jako například jméno nebo příjmení uživatele, jeho příbuzných, hesla odvozená z osobních údajů (například adresa, telefon atd.) nebo jednoduchá slova. Každé heslo by mělo obsahovat víc než 7 znaků a mělo by obsahovat číslice, interpunkci a písmena různé velikosti. Nejlepší hesla jsou nesmyslná slova nebo kombinace jednoduchých krátkých slov, která spolu vzájemně nesouvisejí. Heslo je dobré měnit zhruba každé dva až tři měsíce a pokaždé používat jiné heslo.

Je důležité kontrolovat, jestli mají všichni uživatelé nastavené heslo. Je vhodné si vytvořit vlastní skript, který nás upozorní, pokud uživatel nemá vyplněné heslo. [1][5]

4. Správa systému souborů

V Linuxu se podobně jako v ostatních operačních systémech používá hierarchický souborový systém se stromovou strukturou. Linux používá pouze jeden kořenový adresář, označený '/', podobně jako v operačních systémech vycházejících z Unixu. Další souborové systémy se mohou připojit do kterékoli větve souborového systému.

4.1. Disky

Disky jsou nezbytné pro trvalé uložení dat, neboť data uložená v paměti se po vypnutí ztratí. Disky používají různá rozhraní pro připojení ke sběrnici počítače. Tato rozhraní se převážně liší rychlostí přenosu dat.

Z pohledu přístupu k datům disky patří mezi bloková zařízení s náhodným přístupem.

4.1.1. Disková rozhraní

PATA

Rozhraní PATA neboli IDE je jednoduché a levné rozhraní. Používá sběrnici ATA-7, která má rychlost 133MB/s. Podporuje DMA, což je technologie přímého přístupu do paměti, tzn. bez účasti procesoru. Konektor k připojení zařízení má 39 pinů. Při připojování pevného disku k sběrnici používáme 80žilový kabel a na jednom kabelu mohou být připojeny maximálně dvě zařízení. Maximální délka kabelu je cca 45cm. V případě, že jsou na sběrnici připojena dvě zařízení, jedno zařízení pracuje jako hlavní (master) a druhé jako podřízené (slave). PATA disky jsou vhodné především do pracovních stanic typu PC. Do serverů se příliš nehodí z důvodu absence paralelního zpracování a pomalého přístupu. V současné době je rozhraní PATA na ústupu a je nahrazováno rozhraním SATA. [1]

SATA

Rozhraní SATA je sériové, na rozdíl od rozhraní PATA, které je paralelní. Dosahuje vyšších rychlostí díky pokročilejší technologii, přenos probíhá sériově

na vysoké frekvenci (až 3000 MHz). Nejnovější připravovaná verze SATA 3 by měla mít propustnost 6Gb/s a maximální rychlost přenosu dat 600MB/s. Každý disk se připojuje samostatným kabelem k řadiči a tím pádem se nemusí rozlišovat na master a slave. Oproti PATA podporuje navíc hot swaping což je odpojování a připojování zařízení za chodu počítače a také technologii NCQ. Při použití NCQ pevný disk sám optimalizuje pořadí, ve kterém jsou vykonány požadavky na zápis nebo čtení. Tato optimalizace může redukovat nadbytečný pohyb hlaviček disku. Tím se zvýší rychlost přenosu dat mezi řadičem a diskem a také se mírně sníží opotřebení disku. Konektor pro připojení zařízení má 7 pinů.

Hlavní výhoda rozhraní SATA je jeho cena. Díky vysoké přenosové rychlosti rozhraní SATA představuje levnější alternativu k rozhraní SCSI. [1]

SCSI

Je zkratka pro Small Computer System Interface. SCSI je paralelní rozhraní pro připojení periferií. Používá se zejména pro připojení pevných disků, páskových jednotek nebo optických mechanik. SCSI sběrnice vyžaduje speciální řadič (host adapter), který řídí tuto sběrnici. Na sběrnici může být připojeno několik zařízení, která spolu mohou komunikovat. Každé periferní zařízení má vlastní SCSI rozhraní pro připojení na SCSI sběrnici. Provoz na sběrnici SCSI je zcela oddělen od systémové sběrnice počítače. Každá periferie SCSI je jednoznačně identifikována adresou (ID), která musí být unikátní. SCSI používá detekci chyb pro zvýšení spolehlivosti přenosu dat.

SCSI má několik standardů, SCSI-1. SCSI-2 a SCSI-3, které se hlavně liší šířkou sběrnice, maximální rychlostí přenosu a maximálním počtem připojených zařízení. Na jedné sběrnici může být připojeno až 16 zařízení. Taktovací frekvence sběrnice může být až 160MHz (maximální přenosová rychlost je 320 MB/s).

SCSI je využíváno především tam, kde potřebujeme vyšší výkon, který je zajištěn paralelním zpracováním požadavků a dostatečnou přenosovou kapacitou. Z pohledu využití pro servery je dominantní rozhraní SCSI. [1]

4.1.2. Správa disků

Přidání disku

Přidání nového disku pod operačním systémem Linux je relativně jednoduché. Operační systém nový disk po startu automaticky nalezne a jediným úkolem je tedy disk rozdělit na oddíly a vytvořit na něm souborové systémy. Pokud si nejsme jisti, zdali operační systém disk rozpoznal, je vhodné pomocí příkazu **dmeg** zjistit, jestli je nainstalován příslušný ovladač a zda byl disk skutečně rozpoznán.

V systému Linux vystupuje každý disk jako zařízení s vlastním názvem. Název udává soubor zařízení (angl. device file), který je uložen v adresáři `/dev`. Zařízení typu IDE mají název `hdX` a SCSI nebo SATA, disky mají název `sdX`. X je malé písmeno, které identifikuje fyzické zařízení. [1][5][6]

Rozdělení disku na oddíly

Oddíl je souvislá část disku. Oddíly umožňují zacházet s prostorem na disku jako s nezávislými datovými oblastmi. Každý oddíl je možno použít pro spuštění jednoho operačního systému, který je nainstalován na příslušném oddílu. Umístění a velikost oddílu je uložena v tabulce oddílů (partition table) v prvním sektoru disku. Tento sektor (Master boot record) obsahuje také zaváděcí program (loader), který zajišťuje spuštění OS z aktivního oddílu. Standardní zavaděč umožňuje použít maximálně 4 oddíly na jednom disku.

Linux používá pro každý oddíl nezávislé zařízení. Názvy se tvoří z původního názvu zařízení disku. Například na disku typu IDE s názvem `hda` vytvoříme dva oddíly. Jejich názvy zařízení potom budou `hda1` a `hda2`.

Operační systém Linux je velmi variabilní a může používat jeden nebo několik oddílů. Počet a velikost oddílů je závislá na plánovaném použití. V současné době je pro pracovní stanice oblíbený koncept tří oddílů. Tyto oddíly jsou kořenový oddíl, odkládací oddíl a oddíl s uživatelskými daty (domácími adresáři).

Na prázdném disku si můžeme vytvořit oddíly podle potřeby. Na disku s existujícími oddíly jsme omezeni velikostí volného prostoru (nepřiděleného některému z existujících oddílů).

Pro práci s oddíly se používají různé programy (například **gparted**, **Qtparted**, **parted** atp.). My se však zaměříme na systémový program **fdisk**. Tento program se standardně dodává se všemi distribucemi Linuxu (podobný program se obecně dodává se všemi operačními systémy). Používá rozhraní příkazové řádky, a proto je méně přívětivý než podobné nástroje s grafickým rozhraním.

Nástroj **fdisk** nám umožňuje rušit existující oddíly na disku, vytvářet nové a nastavovat aktivní oddíl (z kterého se bude provádět zavlečení operačního systému). Standardně pracuje s maximálně čtyřmi oddíly na jednom disku. Toto omezení vyplývá z toho, že partiton table má místo jen na čtyři záznamy o oddílech. Alternativně můžeme vytvořit tzv. rozšířený oddíl, což je oddíl vytvořený na již existujícím oddílu. [1][5][7]

4.2. Souborový systém

Hlavním účelem souborového systému je efektivně a bezpečně číst/ukládat data z/do souborů. OpenSUSE 10.3 používá implicitně souborový systém ext3, ale nabízí i jiné souborové systémy, jako například ReiserFS, JFS, XFS, ext2 a síťový souborový systém (NFS). Dále podporuje další souborové systémy, jako například FAT, NTFS apod.

4.2.1. Hlavní typy souborových systémů v Linuxu

Souborový systém ext3

V současnosti je to nejpoužívanější souborový systém v operačním systému Linux. Souborový systém ext3 je rozšířením ext2. Základní struktura souborového systému zůstala stejná, jen došlo k vylepšení spolehlivosti. Spolehlivost je zvýšena použitím žurnálování.

Žurnálování je technika, která zajišťuje integritu dat i v případě výpadku systému (např. při přerušení napájení v průběhu práce). Všechny požadované změny jsou zapisovány do žurnálu jako samostatné transakce. Po fyzickém dokončení každé

transakce se odstraní příslušný záznam ze žurnálu. V případě výpadku je díky žurnálu systém schopen se dostat zpět do konzistentního stavu. Velikost žurnálu a jaké operace budou žurnálovány, lze nastavit.

Další výhodou souborového systému ext3 je například podpora ACL (Access Control Lists) a EA (Extended atributes) nebo správa vadných sektorů a kvót. [1][8]

Souborový systém ReiserFS

Souborový systém ReiserFS dříve používaly jako výchozí všechny verze openSUSE do verze 10.2. ReiserFS podporuje stejně jako ext3 žurnálování a navíc umožňuje lepší definování, jak se mají soubory zpracovat a zabezpečit. ReiserFS je založen na rychlém balancovaném stromu, což nabízí efektivní práci s velkým množstvím maličkých souborů. Dále se snaží spojit výhody souborového systému s výhodami databází. Aby toto bylo možné, ukládají se zvlášť konce (neboli tails) souborů, které nezaplní celý blok. Do jednoho bloku obvykle uloží ocásky více souborů. To na druhou stranu zpomaluje práci se středně velkými soubory, takže se někdy vyplatí filesystém připojit s volbou notail, čímž se tato vlastnost vypne.

Mezi hlavní nevýhody tohoto souborového systému patří absence spolehlivého nástroje na opravu rozpadlých oddílů. To by sice díky žurnálování nemělo teoreticky nikdy nastat, ale v praxi se to stát může. [1][8][9]

Souborový systém XFS

XFS byl vyvinut firmou SGI. V roce 2001 vyšla podpora tohoto souborového systému i pro operační systém Linux. Jedná se o souborový systém, který podporuje žurnálování a má 64bitové adresování. Tento souborový systém se používá především pro servery (obsahuje podporu pro multiprocesorové počítače) a zajišťuje rychlé zotavení po havárii.

Jeho hlavními výhodami jsou velká přenosová rychlost, vysoký výkon a spolehlivost. Má však i svoje nevýhody, jako je slabší výkon při mazání velkého počtu malých souborů, větší náročnost na hardware nebo příliš veliký kód. [1][8][9]

Souborový systém JFS

JFS je 64bitový souborový systém vyvinutý firmou IBM. JFS je pravděpodobně první souborový systém, ve kterém se objevilo žurnálování, dynamické alokování inodů a komprimované uložení alokačních tabulek.

Mezi jeho výhody patří žurnálování, různé velikosti bloků (podporuje bloky o velikostech 512, 1024, 2048, 4096 bytů), což dovoluje zoptimalizovat výkon systému. Dále má podporu řídkých souborů a podporu velkých souborů a souborových systémů. [8][9]

Síťový souborový systém (NFS)

Síťový souborový systém slouží ke sdílení souborových systémů mezi počítači. NFS používá technologii klient – server. Skládá se z několika částí, mezi které mimo jiné patří protokol, server pro připojování svazků, démony řídící základní souborové služby a diagnostické nástroje. NFS je bezestavový, proto každý požadavek na NFS server musí obsahovat všechny informace k provedení procedury. Systém NFS může používat ke spojení protokol TCP nebo UDP. Jaký protokol bude použit, si můžeme zvolit pomocí voleb v příkazu **mount**.

Komunikace mezi serverem a klientem probíhá ve dvou fázích. Nejprve se pomocí protokolu mount připojí adresářový strom k lokálnímu stromu. Během připojování dojde k autentikaci, poté je předán identifikátor kořenového adresáře, který se následně používá při komunikaci protokolem NFS.

Na serveru musí být spuštěni následující démoni – démon na mapování portů (**portmap**), NFS server démon (**nfsd**) a NFS mount démon (**mountd**). Na NFS serveru se konfigurace sdílených souborových systémů provádí v souboru `/etc/exports`. Tento soubor obsahuje seznam svazků, které server poskytuje k připojení, včetně seznamu oprávněných klientů. Soubor `/etc/exports` definuje také přístupová práva pro exportované souborové systémy. Tento soubor je používán démony **mountd** a **nfsd**.

Někdy se vyplatí nepřipojovat svazky ihned po startu systému (například domovské adresáře uživatelů), ale až v případě jejich potřeby. K tomuto účelu slouží démon **automount** (popř. **amd**), který zajišťuje automatické připojování svazku v případě potřeby a odpojení, pokud nejsou určitou dobu používány.

Dále je možné zobrazit různé statistické informace o systému NFS. Pomocí příkazu **nfsstat** můžeme zobrazit statistiku jak serverových procesů, tak statistiku týkající se klientských operací. [5][10][11]

4.2.2. Správa souborového systému

Soubor /etc/fstab

Soubor `/etc/fstab` (příloha C) uchovává seznam souborových systémů, které jsou připojeny do konkrétního systému. Informace v tomto souboru mimo jiné umožňují připojování souborových systémů automaticky při spuštění a kontrolu souborového systému. Dále podává informace o rozložení souborového systému na disku. Právo zapisovat do něj má pouze superuživatel.

Každému řádku v souboru `/etc/fstab` odpovídá jeden souborový systém. V každém řádku je šest polí oddělených mezerami. Pořadí zde hraje roli, protože tento soubor je čten a zpracováván sekvenčně. Platí, že souborový systém na první řádce je připojen před souborovým systémem na druhé řádce atd. Proto musíme dbát na to, aby rodičovské souborové systémy byly připojené před potomky.

První pole popisuje blokové zařízení nebo vzdálený svazek pro připojení. Pro obvyklá připojení to znamená odkaz na název blokového zařízení, které má být připojeno (např. `/dev/hda2`), nebo označení vzdáleného svazku pomocí NFS (např. `server::/home/jirka`). [11]

Druhé pole obsahuje informace o tom, na jaký připojovací bod (adresář) se má daný souborový systém připojit.

Třetí pole udává typ daného souborového systému. Souborové systémy, které podporuje jádro operačního systému, zjistíme v souboru `/proc/filesystems`. Swap označuje svazek nebo soubor použitý jako odkládací prostor.

Do čtvrtého pole se zaznamenávají parametry pro připojení. Mezi hlavní parametry patří, jestli se disk nemá připojovat automaticky po startu, povolení uživatelům použít příkaz **mount** na daný souborový systém nebo jestli se má otevřít souborový systém pouze pro čtení.

Páté pole obsahuje informace, které využívá zálohovací program **dump**. Pokud je v pátém poli hodnota nula nebo není definovaná, znamená to, že daný souborový systém nebude zálohovat.

Šesté pole využívá program **fsck** k určení pořadí pro kontrolu svazků. Pokud je jeho hodnota nula nebo není definována, program **fsck** nebude daný souborový systém kontrolovat. [1][5]

Vytvoření souborového systému

Souborové systémy vytváříme na oddílech. Na jednom fyzickém disku může být i více souborových systémů. Každý souborový systém musí mít v adresáři `/dev` svůj vlastní soubor zařízení (device file).

Souborový systém ext3 na daném diskovém oddílu vytvoříme pomocí příkazu:

```
mke2fs –j
```

Příkazem **mke2fs** můžeme ovlivnit i velikost žurnálovacího souboru, popřípadě specifikovat umístění tohoto souboru na jiném zařízení. [1]

Připojení a odpojení souborového systému

Souborový systém je nutné nejdříve připojit. Připojení souborového systému provedeme pomocí příkazu **mount**. Příkaz **mount** připojí souborový systém na daný adresář v již existující stromové struktuře. Tento adresář se nazývá přípojný bod a je po dobu připojení nepřístupný.

Příkaz **mount** může pro souborové systémy, které nejsou uvedeny v souboru `/etc/fstab`, provádět pouze superuživatel. Ostatní uživatelé mohou používat příkaz **mount** pouze pro souborové systémy, pro které mají `v /etc/fstab` nastaven parametr `users`. Souborové systémy, které jsou uvedeny v souboru `/etc/fstab`, lze připojovat zjednodušeně jen použitím názvu oddílu nebo přípojného bodu.

S mountováním ještě souvisí soubor `/etc/mtab` a `/proc/mounts`. Oba soubory ukazují aktuálně připojené souborové systémy. Soubor `/etc/mtab` bývá někdy jen odkaz na `/proc/mounts`, ale pokud tomu tak není, jde o běžný soubor, do kterého zapisuje příkaz **mount**. Obsah tohoto souboru je závislý na tom, co do něj zapíše **mount** (anebo někdo jiný) a nemusí reprezentovat skutečný stav. Naopak

`/proc/mounts` je soubor, do kterého jádro "za běhu" zapíše používané souborové systémy, takže by v něm měla být situace vždy taková, jak ji vidí jádro. Jistou nevýhodou oproti `/etc/mtab` je to, že jádro neví o všech parametrech předaných příkazu **mount**, takže `/etc/mtab` může obsahovat některé informace navíc. Ale obecně je `/proc/mounts` spolehlivějším zdrojem informací.

V případě připojování vzdáleného svazku pomocí systému NFS musíme mít na straně klienta jádro s podporou NFS mounting. Souborové systémy NFS se připojují podobně jako lokální souborové systémy pomocí příkazu **mount**. V příkazu **mount** ovšem musíme specifikovat jméno serveru, ke kterému se připojujeme, a parametry pro připojení souborového systému.

Odpojování souborových systémů provádíme pomocí příkazu **umount**. Příkaz umount má velmi podobnou syntaxi jako příkaz **mount**. Používaný souborový systém nelze odpojit. Nejdříve se musí zavřít všechny otevřené soubory a ukončit všechny procesy, které používají daný souborový systém. [1][5][11]

Odkládací prostor

Operační systém Linux používá virtuální paměť. Z tohoto důvodu potřebuje odkládací prostor. Jako odkládací prostor (swap space) se používá vyhrazené místo na disku, buď samostatný oddíl nebo speciální soubor určený pro tento účel. Jádro systému v případě nedostatku fyzické paměti ukládá obsah paměťových bloků neaktivních procesů na pevný disk. Tím si uvolňuje místo ve fyzické paměti pro aktivní procesy. V případě potřeby odložených paměťových bloků je jádro znovu načte do fyzické paměti z pevného disku.

Operační systém Linux může používat jako odkládací prostor soubor nebo diskový oddíl. Většinou se pro odkládání používají oddíly. Linux může použít i více odkládacích prostorů (kombinace oddílů a souborů).

Vytvoření odkládacího prostoru na oddílu se provádí příkazem **mkswap**. Aktivace odkládacího prostoru se provede příkazem **swapon**. Odkládací oddíly je vhodné doplnit do souboru `/etc/fstab`, aby se odkládací oddíly připojily automaticky při startu systému. [1]

Kontrola a oprava souborového systému

Současné souborové systémy jsou díky žurnálování relativně odolné proti výpadku systému. Přesto je však potřeba, abychom dbali na udržování disku v bezchybném stavu. Kontrolu a opravu menších chyb zajišťuje nástroj **fsck**. Nástroj **fsck** se většinou spouští při startu operačního systému.

Pokud **fsck** zjistí, že souborový systém byl neregulérně odpojen, snaží se nalézt chyby a opravit je. Nejčastější typy poškození, které nástroj **fsck** opravuje, jsou nepřipojené i-uzly, příliš velký počet odkazů, nezaznamenané nepoužité datové bloky, datové bloky, které jsou evidovány jako volné, ale jsou použité, a neplatné informace v superbloku.

Pokud má souborový systém žurnálování nástroj, **fsck** vrátí žurnál do posledního konzistentního stavu.

Nástroj **fsck** by neměl být pouštěn za běhu operačního systému. Pokud potřebujeme spustit nástroj **fsck** za běhu systému, musíme nejdříve odpojit kontrolovaný souborový systém pomocí příkazu **umount**. [1][5]

4.3. Přístupová práva

Superuživatel má přístup ke všem souborům a procesům bez omezení, nezávisle na nastavení přístupových práv.

Každý soubor nebo proces má svého vlastníka. Vlastník souboru nebo procesu má možnost měnit oprávnění pro daný soubor nebo proces. Vlastník specifikuje, jaká práva budou mít ostatní uživatelé k danému objektu.

Každý soubor, program či spuštěný proces v Linuxu má nastavena přístupová práva. Pro soubory tato práva definují, kdo má oprávnění soubory číst, měnit či spouštět. U spuštěných procesů nebo běžících programů je tím řečeno, kdo má práva tyto procesy zastavovat či s nimi jinak manipulovat. [1][6]

4.3.1. Bity oprávnění

Bitů oprávnění je devět a určují, jaké operace se mohou se soubory provádět a kdo je může provádět. Práva se specifikují pro tři kategorie, a to pro vlastníka, skupinu a ostatní uživatele. Pro každou kategorii jsou rezervovány 3 bity. První bit

definuje právo číst, druhý bit právo zápisu a třetí bit právo spouštění (v případě adresáře tento bit znamená právo čtení z adresáře). Pokud je právo přiděleno, je odpovídající bit 1, pokud přiděleno není, je daný bit nastaven na 0. [6]

4.3.2. Bity setuid a setgid

Bity setuid a setgid jsou nadstavbou klasických 9 bitů oprávnění. Tyto bity umožňují řízené předávání přístupových práv. Oprávnění setuid a setgid má svůj význam hlavně u programů a procesů. Když má program nastavený setuid nebo setgid bit, převezme práva vlastníka (skupiny) programu, nikoliv toho, kdo jej spustil. Toto je využíváno hlavně u systémových programů např. **passwd**, který modifikuje soubor `/etc/shadow`. Soubor `/etc/shadow` je vlastněn superuživatelem a skupinou shadow a bez setuid nebo setgid bitu by nemohl program **passwd** měnit jejich obsah. [1][6]

4.3.3. Sticky bit

Tento bit byl v minulosti důležitý u spustitelných souborů. Pro soubor, který byl spuštěn, si operační systém uchovával některé datové struktury ve virtuální paměti, takže jeho opětovné spuštění bylo rychlejší. V současné době se sticky bit používá u adresářů. Je-li tento bit zapnut, může soubory v daném adresáři smazat nebo přejmenovat pouze vlastník nebo superuživatel. Obvykle se sticky bit nastavuje pro adresář `/tmp`, aby se zabránilo uživatelům mazat nebo přemisťovat soubory ostatních uživatelů. [1]

4.3.4. Změna oprávění

Oprávnění souborů se mění pomocí příkazu **chmod**. Příkaz **chmod** používá při nastavování oprávnění buď osmičkovou syntaxi, nebo symbolický zápis. Osmičková syntaxi se skládá ze tří (respektive 4 pokud nastavujeme bity setuid, setgid, a sticky) číslic – práva vlastníka, skupiny a ostatních uživatelů. Tato syntaxe nastavuje vždy všechny bity oprávnění najednou. Symbolická syntaxe umožňuje nastavovat i jednotlivé bity a neměnit tím ostatní nastavení. Tato syntaxe používá pro nastavení práv symboly r, w, x a znaménko před symbolem. + znamená přidání a – znamená odebrání. Pro označení skupiny práv se používá znaků u, g a o.

U znamená práva vlastníka, g práva skupiny a o práva pro ostatní. Například přidání práva zápisu k souborům pro vlastníka lze jednoduše provést symbolickou volbou „u+w“. Příkaz potom nastaví pouze tento jeden bit a ostatní bity zůstanou nezměněny. [1]

4.3.5. Nastavení implicitního oprávnění

Pomocí příkazu **umask** můžeme ovlivnit implicitní oprávnění, které budou mít nově vytvářené soubory uživatele. Příkaz **umask** používá stejně jako **chmod** osmičkovou syntaxi a číslice představují, která práva se mají odebrat. Například pokud nastavíme **umask** na 111, znamená to, že všechny nově vytvořené soubory budou mít přístupová práva pro všechny jen pro čtení a zapisování. [1]

4.3.6. ACL

ACL umožňuje definovat přístupová práva pro jednotlivého uživatele nebo skupinu. Přístupová práva souboru jsou definována jedním seznamem řízení přístupu (Access control list). Tradiční kategorie vlastník, skupina a ostatní mají stejný význam jako v případě tradičních práv. Tyto tři kategorie jsou v ACL povinné. Tyto záznamy zajišťují zpětnou kompatibilitu s tradičním systémem přístupových práv. Pokud definujeme práva pro konkrétního uživatele nebo skupinu, musíme nastavit přístupovou masku. Přístupová maska definuje omezení přístupových práv konkrétním uživatelům a skupinám specifikovanými záznamy ACL.

Když má při vyhodnocování přístupových práv konkrétní uživatel nebo skupina nastavené požadované právo, provede se logický součin přístupových práv v přístupové masce a práv nalezených v záznamu. Pokud se ve výsledku nachází i požadované přístupové právo, je přístup povolen.

Linux implementuje ACL již dlouhou dobu a lze je používat pro souborové systémy ext2, ext3, JFS, XFS, ReiserFS. Pro různé souborové systémy může mít implementace ACL jistá omezení, jedná se zejména o omezení počtu záznamů v ACL (souborový systém ext3 může mít maximálně 32 položek).

Jeden z hlavních důvodů, proč se ACL používá, je specifičtější možnost nastavení přístupových práv a vyšší míra kompatibility s ostatními operačními

systémy. Konkrétně systém Samba, který převádí ACl mezi systémy Linux a Windows. [1][12]

4.4. Bezpečnost dat

4.4.1. Zálohování

Typy zálohovacích technik

Mezi hlavní zálohovací techniky patří úplná záloha (Full backup), přírůstková (inkrementální) záloha a rozdílová (diferenciální) záloha.

Full backup zálohování ukládá celý oddíl (diskový svazek, skupinu adresářů, nebo jen skupinu souborů). Výhodou je, že nám stačí uchovávat jen poslední zálohu, neboť pokrývá veškeré předchozí. Podstatnou nevýhodou je čas nutný k vytvoření zálohy. Další nevýhodou je spotřeba médií, pokud je potřeba uchovávat starší zálohy. Počet potřebných médií potom neúměrně narůstá.

Inkrementální záloha obsahuje pouze data, která byla změněna od vytvoření poslední zálohy (plné nebo předchozí přírůstkové). Je tedy menší než plná záloha a její tvorba zabere méně času. Protože však neobsahuje všechna data, je pro obnovení nutné mít původní plnou zálohu a všechny následující přírůstkové zálohy.

Hlavní nevýhodou inkrementálního zálohování je potřeba všech záloh od poslední plné zálohy, a pokud se jedna inkrementální záloha poškodí, pak není možno plně obnovit původní stav.

Diferenciální záloha obsahuje jen ta data, která byla změněna od plné zálohy. Na rozdíl od přírůstkové zálohy, rozdílová záloha obsahuje všechny změny od poslední plné zálohy. Pro obnovení původního stavu je potřeba pouze dvou záloh, poslední plné a poslední rozdílové zálohy. Obnova je proto rychlejší než přírůstková, protože nemusí zpracovávat všechny předchozí zálohy. [13]

Zálohovací zařízení a média

Zálohování se provádí na externí média. Pro případ katastrofy bychom měli tyto zálohy na externích médiích uschovávat v jiném místě než v sídle firmy,

například v bezpečnostní schránce v bance nebo u nějaké externí firmy specializující se na tuto činnost.

Optická média CD a DVD se používají zejména pro zálohování menších systémů. Hlavní výhodou zálohování na optická média je jejich cena, která se pohybuje v maximálně desítkách korun za jedno CD/DVD. Samotné zálohování je jednoduché: zvolíme data k zálohování a vypálíme je. Z důvodů maximální životnosti by se měla použít co nejnižší rychlost zápisu. Média pečlivě archivujeme. Životnost a spolehlivost optického média je dána hlavně tím, jak kvalitně je vyrobeno. Pokud máme kvalitní médium, kvalitní zapisovací jednotku a dodržíme pravidla správného skladování (tma, optimální vlhkost, teplota, atd.), mohou zálohy vydržet i několik desítek let. Obecně je ale doporučováno zálohy postupně převádět v pravidelných časových intervalech na nová a nová média a tím prodlužovat životnost a minimalizovat ztrátu dat ze starších CD a DVD. Stejný postup je možné uplatnit i na média BlueRay a HD-DVD.

Další možností je zálohování na flash externí paměťová média připojená prostřednictvím USB portů a FireWire. Hlavní výhody jsou mobilita, cena a univerzálnost připojení. Mezi hlavní nevýhody patří omezený počet zápisů a relativně nízká spolehlivost.

Záloha na magnetická pásková média se používá více než 50 let a stále se jedná o nejspolehlivější a cenově nejefektivnější technologii pro zálohování velkého objemu dat s dlouhodobou datovou ochranou. Hlavní výhodou páskových médií je, že jsou objemově malá, mají velkou kapacitu a snadno se přemísťují. Doba skladovatelnosti páskových médií je až 30 let, což dělá z páskových médií nezávislé médium pro archivaci dat. Mezi hlavní typy páskových médií patří DDS-4, DLT/S-DLT, AIT-4, SAIT-1, VXA a LTO-3. Kapacita páskových médií se pohybuje v rozmezí 20-500 GB. Velké firmy pro usnadnění obsluhy používají pásková média v automatizovaných systémech, jako např. měnič páskových médií, jukebox nebo páskové knihovny.

Zálohování na disk a obnova dat z disku má oproti tradičnímu zálohování na pásku celou řadu výhod. Mezi hlavní výhody patří, že zálohovaná data jsou uložena na připojených discích, takže jsou neustále k dispozici. Toto nám poskytuje možnost rychlé obnovy dat. Nevýhodou zálohování na disk je výrazně vyšší cena

a to, že prostor na pevném disku je konečný, a tudíž musíme staré zálohy mazat, abychom ho mohli použít znovu. [1][14]

Nástroje pro zálohování a obnovu

Při zálohování lze využít velké množství nástrojů. Mezi tradiční nástroje používané v systémech Linux patří programy **dump** a **tar**. Mimo tyto systémové nástroje lze zvolit i jiné softwarové řešení, například systém Bacula. Při výběru nástroje musíme dbát i na výběr zálohovacího média.

Nástroj **dump** přistupuje přímo k souborovému systému a ne jeho prostřednictvím. Nejdříve si souborový systém načte a seřadí i-uzly, takže se pak hlavička nemusí tolik posouvat a čtení je rychlejší. Přímo podporuje inkrementální zálohování a umožňuje lepší možnosti práce s již vytvořenými archivy. Nástroj **dump** nemá omezení na délku cesty, takže hierarchie mohou být libovolně hluboké a dlouhé názvy nedělají problém. **Dump** má také několik omezení, například ho může použít jen superuživatel, každý souborový systém se musí zálohovat individuálně, lze zálohovat jen lokální souborové systémy a neumí zálohovat libovolný souborový systém.

Nástroj **dump** používá úrovně zálohy. V první řadě musíme provést zálohu úrovně 0 což je full backup záloha. Před zálohou je vhodné oddíl, který chceme zálohovat, odpojit nebo připojit pouze ke čtení, aby nedošlo ke změně souborů v době zálohování. Po full backup záloze nástroj **dump** vytvoří záznam v souboru `/etc/dumpdates`. V souboru `/etc/dumpdates` si uchovává historii pro jednotlivé úrovně inkrementální zálohy. Pokud chceme, aby byl daný souborový systém automaticky zálohován, musíme upravit soubor `/etc/fstab`.

Nástroj **tar** byl původně vytvořen pro archivaci souborů. Ale lze ho použít i k zálohování. Načítá několik souborů nebo adresářů a ty následně archivuje do jednoho souboru a posílá na zálohovací médium. Nevýhoda nástroje **tar** je, že některé verze nemusí podporovat neomezenou délku cesty. Toto omezení brání v zálohování hlubokých hierarchií.

Nástroj **restore** obnovuje data zazálohovaná nástrojem **dump**. Program **restore** nabízí více možností, jak obnovit soubory, a to buď plnou obnovu, interaktivní obnovu nebo zotavení z přerušené obnovy. Obnovení dat se provede do adresáře,

z něhož jsme nástroj **restore** spustili, a v případě, že je záloha na více médiích, nástroj **restore** se nejdříve zeptá, z kterého média chceme data obnovit. Nástroj **restore** si uchovává v adresáři dočasný soubor, v němž se ukládají informace o probíhající obnově. Při použití interaktivní obnovy nás nástroj **restore** nechá procházet zálohou a vybrat jednotlivé soubory, které chceme obnovit.

Obnova zálohovaných souborů pomocí nástroje **tar** je snadná. Obnovení je jen otázkou napsání příkazu. Nástroj **tar** umožňuje také obnovovat jednotlivé soubory či adresáře a umožňuje procházení zálohy. Jedinou nevýhodou je, že nástroj **tar** čte zálohu sekvenčně, takže práce s většími objemy dat je pomalá. [1]

4.4.2. Zvýšení spolehlivosti zápisu / čtení

RAID (redundant array of independent disks) je systém používající více fyzických disků, který zajišťuje, že výpadek jednoho disku nezpůsobí ztrátu uložených dat. Tento systém zvyšuje spolehlivost zápisem nadbytečných dat, které umožňují obnovení dat v případě chyby. Dříve se RAID používal výhradně na serverech, protože vyžaduje speciální řadič. Existuje několik úrovní RAID označovaných číslem. Mezi nejpoužívanější patří RAID 1 a 5.

RAID 0 se využívá jen ke zvýšení výkonu. Pracuje se dvěma nebo více disky a data jsou na nich rovnoměrně rozložena. Nepřináší žádnou vyšší míru bezpečnosti, naopak při havárii jednoho z disků jsou ztracena všechna data. Zvýšení výkonu je dáno paralelními operacemi na všech discích, data se rozloží na všechny použité disky.

RAID 1 (neboli zrcadlení) je nejjednodušší a nejběžnější konfigurace. Pracuje nejčastěji se dvěma disky a veškerá data se ukládají na oba disky současně. V případě výpadku jednoho disku tedy nedochází ke ztrátě dat. Přináší i zrychlení čtení dat z disku, protože data můžeme číst z obou disků zároveň. Zápis však probíhá stejnou rychlostí, protože data se zapisují současně na oba disky. Nevýhodou je, že ztrácíme polovinu celkové kapacity disků.

RAID 5 umožňuje, že kromě vlastních dat se na disky ukládá navíc ještě paritní informace tak, aby byl schopen při výpadku jednoho disku vypočítat chybějící data uložená na vadném disku. Pro svoji funkčnost potřebuje minimálně tři pevné disky. RAID 5 je úspornější a ztrácíme vždy kapacitu pouze jednoho disku.

Například při konfiguraci s 5 disky ztrácíme pouze 1/5 kapacity. Čtení dat je rychlejší, protože data jsou uložena na všech discích, naopak při zápisu rychlost klesá, protože je nutné vypočítávat a ukládat navíc paritu.

Pro vytvoření softwarového systému RAID v Linuxu použijeme příkaz **mdadm**. Po vytvoření pole RAID je vhodné vytvořit konfigurační soubor `/etc/mdadm.conf`, který nám zajistí, že při novém zavedení systému bude RAID zapnut. V tomto souboru je také uložena veškerá konfigurace pole RAID. [1][15]

5. Správa software

5.1. Instalační balíčky

Instalační balíček je soubor, který se používá pro instalaci software na OS Linux. Použití instalačního balíčku je pohodlnější, rychlejší a praktičtější než manuální instalace ze zdrojových kódů. Instalační balíčky se spravují programem pro správu balíčků (v OS openSUSE je to **zypper**).

Existují dva typy instalačních balíčků binární (binary package) a zdrojový (source package). Binární balíčky obsahují již zkompilované aplikace. Zdrojový balíček obsahuje zdrojový kód aplikace a ostatní náležitosti, které potřebujeme ke zkompilování aplikace. Pokud chceme aplikaci nainstalovat, musíme nejprve provést její kompilaci.

Správce balíčků pracuje s repositáři. Repositář je seznam, který obsahuje informace o balíčcích a místě, z něhož se dají stáhnout. Repositáře můžou být například na CD/DVD, FTP serverech, webových serverech nebo lokálních či síťových discích. Hlavní výhodou správce balíčků je, že dokáže zjistit, které další balíčky k instalaci daného balíčku potřebujeme. Správce balíčků si udržuje informace o všech nainstalovaných balíčcích a umí je jednotlivě odstraňovat. [16][17]

5.1.1. Formáty instalačních balíčků

V současné době se používají hlavně dva formáty instalačních balíčků. Distribuce Rad Hat, Fedora, SUSE a některé další používají formát RPM a distribuce Debian a Ubuntu používají formát DEB. Existence více balíčkových formátu je jedna

z nevýhod OS Linux. Neexistuje totiž žádný standardní formát instalačních balíčků, který by implementovaly všechny distribuce. Z tohoto důvodu existuje speciální nástroj **alien**, který umožňuje převést formáty balíčků. [18][19]

RPM

Formát balíčků RPM vytvořila firma Rad Hat. Názvy balíčků se v každé distribuci liší, ale většinou je dodržován předpis `jméno-verze software-release.architektura.rpm`. Release je číslo, označující, kolikrát byl z této verze vytvořen balíček. Architektura označuje architekturu, pro niž byl balíček zkompilován (např. i386, i586, x86_64). Pokud je na místě architektury `noarch`, obsahuje balíček skripty či data nezávislá na platformě. Zdrojový balíček má v poli architektura `src`. Nevýhodou balíčků RPM je, že při instalaci můžeme narazit na problém s cyklickými závislostmi. To znamená, že jeden balíček vyžaduje instalaci druhého a druhý vyžaduje instalaci třetího, jenže třetí vyžaduje instalaci prvního. [17][20]

DEB

Formát balíčků DEB byl původně určen pouze pro distribuci Debian. Binární balíčky mají koncovku `.deb` a obsahují dva archívy. Jeden obsahuje informace o balíčku a druhý data, která se při instalaci kopírují do systému. Zdrojové balíčky nemají žádnou speciální koncovku, protože se skládají ze 3 souborů. První je originální zdrojový balíček, který má příponu `orig.tag.gz`, druhý je soubor, který obsahuje informace o balíčku (má příponu `.dsc`) a tzv. záplatu (patch), jíž se upraví originální zdrojový balíček tak, aby z něj bylo možné zkompilovat binární balíček (má příponu `.diff.gz`). Způsob popisování balíčků je následující `jméno_verze software-revision_architektura.deb`. Revision udává číslo revize balíčku. Balíčky formátu DEB nemají problémy s cyklickými závislostmi. [16][20][21]

5.2. Instalace software

Instalovat nový software můžeme pomocí balíčků nebo zdrojových kódů. Nový software instalujeme pod superuživatelským účtem.

5.2.1. Instalace balíčků

Instalace balíčků je v openSUSE 10.3 realizovaná dvěma způsoby. Můžeme použít nástroj **rpm**, nebo systém pro správu balíčků **zypper**. Názorná ukázka instalace, viz příloha D.

Instalace pomocí nástroje rpm

Nástroj **rpm** slouží k instalování, odinstalování, a ověřování balíčků. Také umožňuje prohlížení již nainstalovaných balíčků. Pro instalaci balíčku pomocí **rpm** musíme mít stažený balíček nebo zadat umístění, kde lze daný balíček nalézt. Nástroj **rpm** neřeší závislosti, což znamená, že pokud instalovaný balíček vyžaduje další balíčky, musíme tyto balíčky nejdříve stáhnout.

Ověření balíčku se provádí porovnáním informací o nainstalovaných souborech s informacemi o balíčku v databázi balíčků. Zkoumá se rozdíl velikosti, přístupových práv, vlastnictví atd. [5]

Instalace pomocí systému pro správu balíčků

Zypper je program, který nám umožňuje vyhledávat, instalovat, odebírat nebo aktualizovat instalované balíčky. Také umožňuje přidávání, odstraňování a modifikování repositářů balíčků. Před instalací balíčků je nutné zadat repositáře, ve kterých může program **zypper** hledat. Při instalaci balíčku stačí zadat název balíčku a program **zypper** prohledá dostupné repositáře. Pokud balíček v repositářích nalezne, zkontroluje závislosti a vypíše, které balíčky budou nainstalovány. Po potvrzení, že chceme skutečně balíčky nainstalovat, stáhne balíčky z repositářů a následně je nainstaluje.

5.2.2. Instalace ze zdrojových kódů

Některý software není dostupný v instalačních balíčcích, ale pouze ve zdrojovém kódu. Je vhodné přečíst si přiložený soubor `README`, který obsahuje informace o software a důležité informace potřebné pro instalaci. Nainstalování software ze zdrojového kódu se skládá z několika kroků. Zdrojový kód je většinou zabalen archivátorem **tar,** a proto ho nejdříve musíme rozbalit. Po rozbalení archivu musíme přejít do adresáře, kam jsme archív rozbalili. Rozbalenému zdrojovému

kódu musíme nastavit proměnné prostředí a jiné důležité informace pro kompilaci. Nastavení těchto poměných se provádí příkazem **./configure**. Následně provedeme již vlastní kompilaci zdrojového kódu. Kompilace nastavených souborů se provádí pomocí příkazu **make**. Posledním krokem instalace je spuštění příkazu **make install**, který nainstaluje software. [5][19]

5.3. Aktualizace software

Je důležité udržovat software aktualizovaný. Aktualizace řeší bezpečnostní díry, chyby v software, vylepšují funkce software apod. Aktualizaci software v openSUSE 10.3 obstarává program **zypper**. Ten rozlišuje dva typy aktualizací. Aktualizaci oprav a aktualizaci software.

5.3.1. Aktualizace oprav

Při aktualizaci oprav (záplat) se neinstalují nové verze software, ale pouze opravy nainstalovaného software. Aktualizaci záplat provádí program **zypper** automatizovaně. Stačí pouze použít příkazu **zypper update** a program **zypper** sám vyhledá a nainstaluje dostupné záplaty. Součástí systému openSUSE 10.3 je nástroj na automatickou aktualizaci oprav. Tento nástroj se pouští automaticky při startu systému, a pokud jsou v repositářích nové dostupné aktualizace, zobrazí informace o nových aktualizacích a dotáže se, zdali je chceme nainstalovat.

5.3.2. Aktualizace software

Aktualizace software lze provést pomocí nástroje **rpm** nebo **zypper**. Aktualizace pomocí **rpm** je stejná jako instalace balíčku, tímto nástrojem jen použijeme jiné parametry příkazu. Pokud chceme aktualizovat software pomocí nástroje **zypper**, postupujeme jako při instalaci balíčku. Zadáme název software, který chceme aktualizovat. **Zypper** prohledá dostupné repositáře, a pokud najde novější verzi, zeptá se, jestli chceme novější verzi nainstalovat.

6. Síťové služby

6.1. Nastavení TCP/IP komunikace

Linux podporuje více síťových rozhraní. Síťové karty jsou detekovány při startu systému. Pro nalezené síťové karty jsou zavlečeny ovladače a rozhraní jsou automaticky přidána do systému a pojmenována např. "`eth0`".

Pro připojení počítače do sítě můžeme použít dva typy nastavení, a to statické nebo dynamické. Statické nastavení se provádí použitím příkazů nebo přidáním záznamů do systémových souborů. Dynamické nastavení používá DHCP (Dynamic Host Configuration Protocol), který umožňuje automaticky nastavit všechny potřebné konfigurační parametry pro síťovou komunikaci (IP adresu atd.).

6.1.1. Statické nastavení

Statické nastavení provádí ručně superuživatel a spočívá v nastavení IP adresy, masky a výchozí brány na pevné hodnoty, které se nemění. Toto nastavení je nevhodné ve větších sítích, protože každé zařízení v síti se musí nastavit ručně, a pokud je potřeba udělat změnu (například změna IP adresy výchozí brány), musí se změnit nastavení všech zařízení v síti.

Nastavení IP adresy a masky

Nastavení IP adresy a masky lze provést pomocí příkazu **ifconfig** nebo upravením konfiguračního souboru. Mimo příkazu **ifconfig** lze také použít komplexnější nástroj **ip**. Tento nástroj však nemusí být k dispozici v každé distribuci, proto se zaměříme na příkaz **ifconfig**, který je dostupný v každém operačním systému Linux.

Pro nastavení IP adresy pomocí **ifconfig** musíme specifikovat argumenty. Prvním argumentem musí být název konfigurovaného síťového zařízení (obvykle eth0) a následně IP adresa, kterou má používat dané síťové zařízení. Specifikace masky sítě se provádí pomocí parametru netmask. Příkaz **ifconfig** nám dále umožňuje zapnout či vypnout síťové zařízení nebo nastavit velikost MTU (Maximum Transfer Unit). MTU je maximální délka paketu přenášená síťovým zařízením.

Konfigurační soubor síťového zařízení (viz obrázek) je v openSUSE 10.3 uložen v adresáři `/etc/sysconfig/network`. Má název `ifcfg-název_zařízení`. Pro upravení IP adresy nás zajímá hlavně řádek IPADDR. Sem vypíšeme IP adresu daného síťového rozhraní. Masku můžeme specifikovat v řádku NETMASK nebo specifikovat pomocí `/počet_bitů` v řádku IPADDR (viz obrázek 1). Další důležité položky tohoto souboru jsou BOOTPROTO, což je položka, určující, jakým způsobem se nastavuje rozhraní a STARTMODE, který definuje, jakým způsobem se dané síťové zařízení zapne. [1][5]

```
Linux:/etc/sysconfig/network # cat ifcfg-eth1
BOOTPROTO='static'
STARTMODE='auto'
NAME='AMD PCnet - Fast 79C971'
BROADCAST=''
ETHTOOL_OPTIONS=''
IPADDR='192.168.1.15/24'
MTU=''
NETMASK=''
NETWORK=''
REMOTE_IPADDR=''
USERCONTROL='no'
```

Obrázek 1: Konfigurační soubor síťového rozhraní

Nastavení výchozí brány

Výchozí brána je adresa směrovače, na který se posílají všechny pakety, pro něž nebyla nalezena adresa směrovače ve směrovací tabulce. Směrovací tabulka obsahuje adresy směrovačů pro cílové sítě společně s adresou síťového zařízení, které se použije pro odeslání paketů. Nastavení výchozí brány se provádí pomocí příkazu **route**. Příkaz **route** slouží k údržbě routovací tabulky. Může přidat, změnit nebo odebrat záznam ve směrovací tabulce. [5][6]

6.1.2. Dynamické nastavení

Dynamické nastavení se provádí pomocí DHCP protokolu. DHCP protokol automaticky nastavuje síťové konfigurační parametry, jako například IP adresu, masku, doménu, výchozí bránu a jmenné DNS servery. DHCP server přiděluje síťové konfigurační parametry na základě požadavku klienta. Klient posílá požadavek při startu systému a žádá o obnovu před vypršením doby platnosti informací. [1]

Nastavení DHCP klienta

Ke komunikaci s DHCP serverem se používá DHCP klient. Instalace a konfigurace je závislá na operačním systému klienta. Pro LINUX se používá démon **dhcpcd**, který se spouští při startu operačního systému. Nejdříve musíme nastavit síťové zařízení, s nímž bude DHCP server pracovat. Upravíme soubor `/etc/sysconfig/network/ifcfg-xxx` (kde `xxx` je název síťového zařízení). V řádku BOOTPROTO nastavíme hodnotu `dhcp`. Při následujícím startu systému by měl již démon **dhcpcd** nastavit konfigurační informace pro síťové zařízení. V konfiguračním souboru /etc/sysconfig/network/dhcp lze specifikovat další nastavení. [5]

6.2. DNS

DNS (Domain Name System) slouží pro použití jmenného hierarchického systému v sítích TCP/IP. Primárně se používá pro překlad doménového jména na IP adresu, ale obecně může poskytovat další informace svázané s doménovým jménem. DNS systém používá stromovou strukturu jmen oddělených tečkami. Každá úroveň tohoto stromu obsahuje informace o příslušné úrovni (doméně). Jejím kořenem jsou kořenové servery, které mají označení ".". Kořenové servery uchovávají informace o všech doménách druhé úrovně (například .com, .edu). Server v doméně druhé úrovně ví o serverech třetí úrovně dané domény (například news.com, ibm.com) a tak dále.

Existují tři druhy jmenných serverů. Primární jmenné servery, na kterých jsou uložena veškerá data příslušné domény, jsou označovány jako autoritativní. Sekundární jmenné servery udržují kopií data z primárních serverů pro příslušné domény. Sekundární servery si automaticky aktualizují informace z primárních serverů buď na základě konfiguračních údajů v doméně, nebo na základě upozornění z primárního serveru, že došlo ke změně údajů v doméně. Třetím druhem jmenných serverů jsou pomocné servery. Ty neobsahují žádné konfigurační soubory, pouze si ukládají odpovědi na dotazy, které jimi prošly.

Dále se jmenné servery rozdělují na rekurzivní a nerekurzivní. Nerekurzivní servery vrací odpověď jen v případě, že má dotaz uložený v cache paměti nebo když

je autoritativní pro danou doménu. Jinak pošle jen odkaz na autoritativní server jiné domény.

Mechanismus vyřizování dotazů rekurzivních serverů funguje na základě delegování. Klient požádá jmenný server o vyhledání IP adresy počítače. Jmenný server v případě, že již stejný požadavek vyřizoval, má IP adresu uloženou ve vyrovnávací paměti, takže může rovnou vyřídit požadavek. V případě, že informaci nemá, postupuje následujícím způsobem. Nejdříve se jmenný server zeptá kořenového serveru. Kořenový server může však odkázat pouze na jmenný server druhé úrovně, takže se lokální server zeptá konkrétního jmenného serveru druhé úrovně. Postup se opakuje do doby, než jmenný server zjistí adresu daného počítače a tu předá klientovi. Adresu si uloží v paměti cache pro případné budoucí použití.

DNS umožňuje i zpětný překlad, což je převod IP adresy na jmennou adresu pomocí speciální domény in-addr.arpa. [1][5][22]

6.2.1. Nastavení DNS klienta

Každý klient, který chce využívat službu DNS musí mít DNS resolver. Resolver se stará o vyřizování požadavků na zjištění IP adresy ze symbolické doménové adresy. Nastavení resolveru závisí na tom, jestli máme k dispozici jmenný server. Pokud není jmenný server k dispozici, musíme nastavit statický seznam adres v souboru `/etc/hosts`. V případě, že máme k dispozici jmenný server, musíme upravit soubor `/etc/resolv.conf`. V obou případech je nutné upravit soubor `/etc/nsswitch.conf`, který říká resolveru, jakým způsobem bude překlad jmen probíhat. Pokud máme k dispozici DHCP server, nemusíme se o nastavení jmenného serveru starat, protože DHCP server předává i informace o DNS serverech.

Soubor `/etc/hosts` obsahuje statický seznam IP adres a jmen hostitelů (případně jejich aliasů). Každý řádek v tomto souboru udává jednu IP adresu hostitele. Například pro navštívení webové stránky www.seznam.cz musíme zjistit IP adresu serveru www.seznam.cz a tu pak zapsat do souboru `/etc/hosts`.

V případě, že máme k dispozici DNS server, musíme upravit soubor `/etc/resolv.conf`. Tento soubor obsahuje informace o dostupných DNS

serverech a může obsahovat maximálně 3 jmenné servery. Prvním údajem je prohledávaná doména. Tento údaj udává, že při vyhledávání požadovaného názvu, který nemá uvedenu doménu, se automaticky postupně připojují názvy těchto domén. Například pokud bude hodnota prohledávané domény nastavena na seznam.cz a my se chtěli připojit k počítači uctarna, nejdříve se zkusí vyhledat počítač uctarna.seznam.cz. Dalšími řádky jsou IP adresy DNS serverů. Mimo tyto údaje se zde může ještě objevit údaj domain, options nebo sortlist.

V souboru `/etc/nsswitch.conf` se také nastavuje, jakým způsobem (službou) a v jakém pořadí má resolver překládat adresy. Toto nastavení se provádí v řádku hosts, kde vypíšeme, jaké služby se mají používat. Například `hosts = dns, files` znamená, že první se použije systém DNS a v případě neúspěchu se prohledá soubor /etc/hosts. [1][5][23]

6.2.2. Nastavení DNS serveru (BIND 9.4.1)

Program BIND má tři součásti. Démon **named**, který řeší dotazy, knihovní rutiny, které řeší hostitelské dotazy kontaktováním serverů distribuované databáze DNS, a nástroje pro správu DNS. Konfigurace **named** se skládá z nastavení souboru `/etc/named.conf` a vytvoření souborů s konfigurací příslušných domén.

Soubor /etc/named.conf

Soubor `/etc/named.conf` specifikuje veškeré nastavení démona **named**. Specifikuje proměnné pro provoz démona **named**, jmenného serveru a zón. Tento soubor se skládá ze skupin příkazů. Každý příkaz musí být ukončen středníkem. Skupina příkazů začíná klíčovým slovem, které označuje typ příkazu. Hlavní klíčová slova jsou `include`, `options`, `acl`, `key`, `trusted-keys`, `server`, `masters`, `logging`, `zone`, `view` a `controls`. Každý z těchto typů příkazů má svoji specifickou syntaxi.

Příkaz `include` umožňuje připojení souborů do `/etc/named.conf`. Používá se zejména pro zkrácení délky souboru a lepší organizaci.

Příkaz `options` specifikuje globální proměnné. Například proměnná `directory` specifikuje adresář, do kterého se budou ukládat výstupní soubory. Pokud se v konfiguračním souboru objeví relativní cesta, bude doplněna hodnotou

této proměnné. Proměnná `notify` specifikuje, jestli je démon **named** hlavním serverem pro dané zóny. Proměnnou `forwarders` nastavíme tzv. předávače. V případě, že DNS server nenajde ve svých datech odpověď na dotaz, posílá dotaz předávači. Proměnnou `recursion` specifikujeme, jestli démon **named** bude požadavky na překlad vyřizovat rekurzivně, či nikoliv.

Příkaz `acl` určuje přístupová práva k jmennému serveru. Protože se soubor `/etc/named.conf` čte sekvenčně, musí být deklarace této proměnné prvním příkazem v souboru.

Příkaz `key` definuje šifrovací klíč, který se bude používat pro ověřování a odesílání požadavků konkrétnímu serveru.

Příkaz `server` nám umožňuje nakonfigurovat specifické údaje o dalších jmenných serverech, s nimiž se démon **named** může setkat.

Příkaz `masters` nám umožňuje používat jméno pro skupinu serverů. Musíme ale specifikovat jejich IP adresy a klíče. Výhodou je, že nemusíme používat IP adresy a klíče, ale stačí se pouze odkazovat přiřazeným názvem.

Příkaz `logging` umožňuje zadat údaje, které se mají zaznamenávat do log souborů. Implicitně se do logů ukládá řada užitečných údajů, jež můžeme najít v souboru `/var/log/messages`.

Příkaz `zone` je hlavní příkaz souboru `/etc/named.conf`. Pomocí tohoto příkazu specifikujeme parametry zóny (domény). Je to například typ zóny, cesta ke konfiguračnímu souboru, nebo seznam sekundárních serverů, které si mohou stahovat informace o zóně. Typy zón jsou například primární, sekundární, hint a forward. Hint zóna odkazuje na soubor, kde jsou uloženy DNS záznamy o kořenových serverech. Forward zóna nahrazuje předávací nastavení démona **named** pro konkrétní doménu.

Příkaz `controls` udává, jak je řízen démon **named** programem **rndc**. Pomocí tohoto příkazu specifikujeme například, komu bude povolen vzdálený přístup pomocí **rndc**, a popřípadě můžeme specifikovat ověřovací klíče. [1][5][24][25]

Vytvoření databáze DNS

Databáze DNS pro určitou doménu obsahuje více souborů (tzv. zónové soubory), které jsou uloženy na primárním jmenném serveru domény. Pro každou zónu existuje jeden zónový soubor. Zónové soubory obsahují dva typy záznamů – záznamy pro analyzátor a zdrojové záznamy. Záznamy pro analyzátor jsou uváděny na začátku souboru. Zdrojové záznamy dělíme na zónové, základní, bezpečnostní a volitelné.

Mezi zónové záznamy patří například záznam SOA a NS. Záznam SOA (start of authority) určuje jmenný server, který je autoritativním zdrojem informací pro danou doménu. Záznam SOA je vždy právě jeden a zpravidla na začátku souboru. NS (name server) záznam definuje, které jmenné servery jsou autoritativní pro danou zónu.

Mezi základní záznamy se řadí například A, PTR a MX záznamy. A (adress) záznamy jsou základní stavební kámen databázových souborů, protože slouží k převodu názvů na IP adresy. Typ záznamu PTR (pointer) slouží k zpětnému překladu IP adresy na jméno. Záznam MX (mail Exchange) specifikuje informace o serveru, který přijímá poštu pro danou zónu.

Bezpečnostní záznamy jsou například záznamy typu DS, DNKEY, NSEC a RRSIG a slouží k ověření původu údajů o zónách a k ověření jejich integrity pomocí veřejného klíče. Tyto záznamy musí být vytvořeny pomocí speciálních programů.

Mezi volitelné záznamy patří například CNAME nebo TXT záznamy. CNAME (canonical name) záznam nám umožňuje vytvořit alias pro název jakéhokoliv počítače. Záznam TXT nám umožňuje přidat libovolný text. [1][5][26]

Nástroje pro správu DNS

Existuje několik programů, které pomáhají a usnadňují údržbu nebo řešení problémů s DNS. Mezi tyto programy patří například **rndc** a **dig**. Dále jsou k dispozici programy **named-checkconf** a **named-checkzone** které kontrolují syntaxi souboru `/etc/named.conf` a zónových souborů.

Program **rndc** umožňuje vzdálený přístup k démonovi **named**. Umožňuje například vypnutí a zapnutí démona **named**, výpis jeho stavu nebo přepnutí do debug

módu. Pokud se používá vzdáleného ovládání, je nutné dobře zabezpečit přístup k této službě.

Nástroj **dig** nám umožňuje zjišťovat nejrůznější informace o jmenných serverech. Například zjištění IP adresy podle názvu počítače či naopak vypsání zdrojových záznamů jmenného serveru. [1][5]

6.3. DHCP server

DHCP server přiděluje klientům síťové konfigurační parametry. Tyto parametry mohou být přidělovány staticky nebo dynamicky.

V případě statického přidělování má DHCP server seznam MAC adres a ke každé této adrese jsou přiděleny konkrétní síťové konfigurační parametry.

Dynamické přidělování funguje na základě přidělování údajů z určitého rozsahu hodnot a není vázáno k MAC adrese. DHCP server poskytuje tyto parametry na předem stanovenou dobu (tzv. lease time) a klient si musí požádat o obnovení přidělených parametrů před vypršením této doby.

Protokol DHCP je založen na všesměrovém vysílání (broadcast), proto musí být DHCP server v každé síti (podsíti) nebo musíme použít DCHP předávač (relay), který umožňuje předávání požadavků na DHCP server v jiné síti. [1][5]

6.3.1. Nastavení DHCP serveru (ISC 3.0.6)

DHCP server je realizován pomocí démona **dhcpd**. Hlavním konfiguračním souborem je `/etc/dhcpd.conf`. V tomto souboru lze nastavit, jaké síťové konfigurační parametry bude démon **dhcpd** přiřazovat. Dále je zde možno specifikovat, na jak dlouho se budou parametry poskytovat a komu je bude přiřazovat. Všechny přiřazené parametry zaznamenává démon **dhcpd** do souboru `/var/lib/dhcp/db/dhcpd.leases`. [1][5]

6.4. WWW server

Systém World Wide Web (WWW) je založen na přenosu souborů pomocí HTTP (Hyper-Text Transfer Protocol) protokolu. Protokol HTTP je založen na architektuře klient – server. Webový klient zašle požadavek serveru. Tento

požadavek má záhlaví, které obsahuje informace o klientovi. Server zpracuje požadavek klienta a pošle požadovaný obsah, který obsahuje záhlaví serveru. Obsah, který vrací server, může být statický nebo dynamický. Statický obsah je načten a beze změny odeslán klientovi. Dynamický obsah server vytvoří pomocí aplikací, např. díky rozhraní CGI (Common Gateway Interface), které serveru umožňuje komunikovat s jinými programy.

WWW servery mohou používat systém virtuálních serverů jež umožňuji aby na jednom fyzickém serveru bylo více webových serverů. Rozlišují se podle IP adresy, komunikačního portu a podle doménového jména. [1][5]

6.4.1. Nastavení WWW serveru (Apache 2.2.4)

Adresář `/etc/apache2` obsahuje konfigurační soubory pro nastavení démona **httpd**. Hlavní konfigurační soubor je `httpd.conf`. Nastavení jednotlivých proměnných je realizováno v samostatných konfiguračních souborech, které jsou připojeny do souboru `http.conf` pomocí příkazu `include`.

Soubor `http.conf` je rozčleněn do tří částí. První část obsahuje globální nastavení, například pod jakým uživatelem bude démon **httpd** běžet, na jakém portu bude démon poslouchat, jaké moduly se mají načíst, nastavení pro chybová hlášení nebo nastavení informací, které se mají logovat.

Druhá část se týká nastavení výchozího serveru. Je zde například specifikován adresář, ve kterém jsou uloženy webové stránky, přístupová práva k těmto souborům, aliasy a informace o výchozím serveru.

Třetí část se týká nastavení virtuálních serverů. V této části se provádí definování virtuálních serverů. Pokud chceme rozlišovat servery pomocí IP adres, musíme nejdříve nastavit virtuální síťové rozhraní pomocí příkazu **ifconfig**. [1][5]

6.5. Elektronická pošta

Systém elektronické pošty je založen na čtyřech hlavních komponentech. MUA (Mail User Agent, uživatelský poštovní program), MTA (Mail Transfer Agent, program pro přenos pošty), MDA (Mail Delivery Agent, program pro doručení pošty) a AA (access agent, přístupový program).

MUA umožňuje uživatelům vytvářet a číst elektronickou poštu. Při odesílání pošty MUA předá obsah zprávy (text, adresát a přidaná základní hlavička) programu MTA.

MTA se stará o přenos pošty mezi počítači. Po převzetí zprávy od MUA doplní hlavičku zprávy o další údaje. Z adresy určí, který počítač je zodpovědný za přijímání pošty pro danou doménu. Následně naváže s tímto počítačem spojení a zprávu mu odešle. Odesílání se realizuje pomocí SMTP (Simple Mail Transfer Protocol) protokolu. MTA na příchozím počítači předá zprávu MDA.

MDA se stará o doručení zpráv uživatelům do jejich datových souborů (poštovních schránek).

AA umožňuje vzdálené připojení uživatelského programu k poštovní schránce na serveru. Zajišťuje přenos pošty, který se provádí pomocí protokolů POP (Post Office Protocol) nebo IMAP (Internet Message Access Protocol). [1][27]

6.5.1. Poštovní server (Postfix 2.4.5)

Program **Postfix** se skládá z několika menších spolupracujících programů. Tyto programy spolu vzájemně komunikují a zajišťují funkčnost poštovního systému (příjímání a odesílání zpráv). Všechny programy jsou spouštěny a řízeny hlavním programem master. [28]

Příjem pošty

Pošta se odesílá a přijímá prostřednictvím démona **smtpd**. Tento démon kontroluje, zda zpráva splňuje základní SMTP pravidla. Dále kontroluje, zda je pošta určena pro jeho doménu, server a jestli existuje uživatel, pro kterého je mail určen. Po kontrole a přenosu předá zprávu programu **cleanup**. Program **cleanup** se stará o přidání chybějící hlavičky a kontrolu obsahu hlavičky. Dále se zpráva předá programu **trivial-rewrite**, který má na starosti ověření adresy a její případnou změnu. Po úpravách adresy **trivial-rewrite** vrátí poštu programu **cleanup**, který poštu zařadí do fronty příchozí pošty a upozorní program spravující tuto frontu na přidání nové pošty. [28]

Doručení pošty

O doručování pošty se stará správce fronty **qmgr**, jenž udržuje několik front: frontu příchozí pošty, právě doručované pošty a frontu pro poštu, která nemůže být z nějakého důvodu doručena. Program **qmgr** používá program **trivial-rewrite** pro rozhodnutí, kam se má pošta poslat. Pro doručení pošty správce fronty kontaktuje jednoho z doručovacích programů. Tyto programy jsou **smtp**, **lmtp**, **local**, **virtual**, **pipe**, **discard** a **error**. [28]

Nastavení poštovního serveru

Hlavní konfigurační soubor programu Postfix je `etc/postfix/main.cf`. Dalším důležitým souborem je `/etc/postfix/master.cf`, ve kterém se konfigurují jednotlivé procesy, jež program master obsluhuje.

V souboru `/etc/postfix/main.cf` lze nastavit cesty k ostatním součástem programu **postfix**, síťové informace a informace o databázi aliasů. Mezi nejdůležitější nastavení patří `myhostname, myorigin a mynetworks`.

Poštovní schránky můžeme specifikovat v případě, že uživatelé mají na serveru vlastní uživatelský účet (systémový) v souboru `/etc/aliases` nebo můžeme použít virtuální poštovní schránky. Virtuální schránky definujeme v souboru `/etc/postfix/virtual` a v souboru `/etc/postfix/vmailbox` specifikujeme adresář uložení pošty. [1][5]

6.6. SSH

Služba SSH (Secure Shell) umožňuje zabezpečený vzdálený přístup a přenos souborů. SSH používá šifrovanou komunikaci pomocí šifrovacích klíčů.

Mechanismus je následující. Nejdříve je vygenerována dvojice klíčů - privátní a veřejný. Veřejný klíč je volně přístupný a jeho pomocí lze zprávu pouze zašifrovat. Privátní klíč slouží k rozšifrování zprávy, která byla zašifrována veřejným klíčem. Dešifrování se provádí lokálně, takže se privátní klíč nikdy neposílá sítí. Pokud se klient chce přihlásit na server, musí mít server k dispozici jeho veřejný klíč. Server tímto veřejným klíčem zašifruje náhodná data a pošle je zpět. Klient výzvu dešifruje pomocí privátního klíče a dešifrovanou ji pošle serveru. Pokud jsou

náhodná data správně rozšifrována, je klientovi povolen přístup k serveru. Existuje několik metod používajících tento způsob ověření.

První možností je, že SSH server zkontroluje, zda je veřejný klíč vzdáleného počítače uložen v lokálním souboru, a pokud ano, pošle vzdálenému počítači k rozšifrování náhodná data. Pokud jsou tato data rozšifrována správně, je uživatel připojen do systému bez výzvy k zadání hesla.

Druhá možnost je použít veřejný klíč k identifikaci uživatele. Pokud se chce uživatel přihlásit do systému, musí mít k dispozici svůj privátní klíč (který bývá většinou zaheslovaný).

Třetí možnost je bez využití klíčů. Vzdálený uživatel se přihlašuje standardním způsobem pomocí uživatelského jména a hesla, které má definováno na serveru. [1][5]

6.6.1. Nastavení SSH serveru (openSSH 4.6)

Úložiště veřejných klíčů je v adresáři `/etc/ssh`. Zde se také nachází konfigurační soubor serverového démona **sshd**. V souboru `/etc/ssh/sshd.conf` můžeme například specifikovat způsob komunikace, zakázat přihlášení jednotlivým uživatelům či skupinám a nebo zakázat přihlášení pod účtem root. [5]

6.7. Samba

Samba je balík programů umožňující komunikaci a výměnu dat mezi operačními systémy Linux a Microsoft Windows. Samba implementuje protokol CIFS (Common Internet File System), kterým rovněž komunikují operační systémy Microsoft. To umožňuje přistupovat z operačního systému Linux do sítě založené na MS Windows a naopak. Samba například nabízí možnost sdílení souborů a tiskáren, síťový tisk, autentikaci a autorizaci, vyhledávání jmen a může provádět všechny základní funkce primárního serveru Windows domény.

Většina funkcí Samby je poskytována démony **smbd** a **nmbd**. Démon **smbd** obstarává veškerou TCP/IP komunikaci, autentikaci a sdílení souborových systémů a tiskových služeb. Tento démon naslouchá na portu 139 a čeká na požadavky.

Pokud se uživatel přihlásí, **smbd** démon spustí svoji novou kopii, která dostane přístupová práva uživatele, a potom obsluhuje požadavky uživatele.

Démon **nmbd** zajišťuje vyhledávání jmen a ohlašování služeb. Lze ho také nastavit jako WINS (Windows Internet Name Server) server. To znamená, že **nmbd** bude odpovídat na příslušné dotazy této služby.

Díky odlišnému způsobu šifrování hesel v operačních systémech Windows nelze použít k ověření soubor `/etc/shadow`. Proto si systém samba musí vytvořit vlastní soubor se zašifrovanými hesly. Uživatele a hesla do tohoto souboru přidává superuživatel pomocí příkazu **smbpasswd**. Tento příkaz také umožňuje uživatelům změnit si jejich heslo pro přístup do systému samba. [1][5][29]

6.7.1. Nastavení samba serveru

Nastavení samba serveru lze provést dvěma způsoby. Můžeme upravit konfigurační soubor `/etc/samba/smb.conf` nebo použít konfigurační nástroj SWAT (Samba Web Administration Tool).

Soubor `smb.conf` se skládá ze čtyř částí. Část global určuje nastavení samba systému, část homes určuje nastavení sdílení domácích adresářů uživatelů, část printers nastavuje sdílené tiskárny a poslední částí je nastavení ostatních sdílených adresářů.

Konfigurační nástroj SWAT je webový administrativní nástroj Samby. Umožňuje správu z prostředí webového prohlížeče. Toto rozhraní umožňuje snadnější a přehlednější konfiguraci celého Samba systému. Dále umožňuje správu účtů, které jsou definovány v systému Samba, nastavení sdílených adresářů a tiskáren a poskytuje informace o stavu démonů a připojených uživatelích. [1][5]

7. Praktické nastaveni a administrace domácího serveru

Dále popisované nastavení domácího serveru předpokládá využití serveru pouze několika uživateli. Tento server bude zajišťovat převážně služby pro lokální síť. Předpokládá se externí internetové připojení s externím firewallem a routrem (například ADSL modem). Tento server bude poskytovat následující služby:

- DHCP
- DNS (lokální + internet)
- souborový server
- webový server
- poštovní server

7.1. Instalace a konfigurace operačního systému

V této kapitole probereme instalaci a konfiguraci operačního systému openSUSE 10.3.

7.1.1. Instalace

Při instalaci nakonfigurujeme síťové rozhraní podle parametrů lokální sítě a pro přístup k internetu. Nastavíme IP adresu, síťovou masku, výchozí bránu nastavenu na router, který je připojen do internetu, a DNS server poskytovatele připojení. Použil jsem následující nastavení: IP adresu serveru 192.168.0.2, masku 255.255.255.0 a výchozí bránu 192.168.0.1.

Server nevyžaduje instalaci grafického prostředí. Přesto jej nainstalujeme pro usnadnění administrace pomocí grafických nástrojů. Jako grafické prostředí jsem zvolil KDE, protože se domnívám, že je přehledné a snadno se ovládá.

Při rozdělování disku jsem zvolil model tří oddílů. Vytvořil jsem odkládací oddíl o velikosti 10GB, kořenový oddíl o velikosti 20GB a oddíl pro domácí adresáře o velikosti 130GB.

Při instalaci software jsem zvolil základní systém a vybraný serverový software. Software pro pracovní stanice jsem z důvodů úspory místa neinstaloval.

V konfiguraci sítě zakážeme podporu protokolu IPv6 a vypneme firewall. Dále zvolíme jméno počítače a lokální domény.

Po nakonfigurování sítě instalační program vyhledá dostupné aktualizace pro operační systém a nainstalovaný software.

Po dokončení instalace již máme plně funkční operační systém a můžeme začít s nastavením služeb.

7.1.2. Instalovaný software

- Základní operační systém openSUSE 10.3
- DHCP 3.0.6 (dhcp-server-3.0.6-24.2.i586.rpm)
- BIND 9.4.1 (bind-9.4.1.P1-12.7.i586.rpm)
- Samba 3.0.26 (samba-3.0.26a-3.7.i586.rpm)
- Apache2 2.2.4 (apache2-2.2.4-70.8.i586.rpm)
- Postfix 2.4.5 (postfix-2.4.5-2.0.6.i586.rpm)
- Dovecot 1.0.5 (dovecot-1.0.5-6.4.i586.rpm)

7.1.3. Instalace balíčků

Instalaci uvedených balíčků provedeme pomocí programu zypper. Balíčky nainstalujeme z hlavního repositáře openSUSE 10.3. Například stažení a instalaci balíčku pro program BIND provedeme následujícím příkazem.

```
zypper install bind
```

Pro instalaci dalších balíčků bychom postupovali stejným způsobem.

7.1.4. Konfigurace systému

Systém používá standardní nastavení, nevyžaduje žádné specifické nastavení.

7.2. Nastavení DHCP

Nejprve musíme stanovit rozsah dynamicky přidělovaných IP adres pro lokální síť. Tuto konfiguraci provedeme v souboru `/etc/dhcpd.conf`.

Zde musíme stanovit síť, pro kterou má server přidělovat síťové parametry. Dále musíme stanovit, jaké parametry bude server přidělovat. To provedeme následujícím zápisem:

```
Subnet 192.168.0.0 netmask 255.255.255.0 {
      option domain-name "home";    #nastavení domény
      option domain-name-servers 192.168.0.2;
                                    #nastavení DNS serveru
      option routers 192.168.0.1;   # IP adresa routeru
      range 192.168.0.129 192.168.0.254;
                                    #rozsah přidělovaných IP adres
      default-lease-time 14400;     #hodnota v sekundách
      max-lease-time 172800;
}
```

DHCP server bude tedy přidělovat doménu, adresu DNS serveru, výchozí bránu, IP adresy v daném rozsahu a maximální doba zapůjčení jsou dva dny.

Automatické spuštění DHCP serveru po startu systému zajistíme vytvořením symbolických odkazů na startovací skript v adresáři `/etc/init.d/dhcpd`. Tyto symbolické odkazy vytváříme v `/etc/rc.d/rcX.d`, kde `X` je číslo úrovně běhu systému. Symbolický link se musí jmenovat `Snndhcpd` pro start a `Knndhcpd` pro ukončení, `nn` je pořadí provedení skriptu.

7.3. Nastavení DNS

K nastavení lokálního DNS serveru je potřeba znát doménové jméno. Dále je potřeba znát IP adresu DNS serveru v internetu pro resoluci internetových informací. Nejprve provedeme konfiguraci souboru `/etc/named.conf`. Zde musíme nastavit předávače a vytvořit zóny. Předávače nastavíme v části options přidáním řádku

```
      forwarders { 212.158.128.2; 212.158.128.3; };
      #DNS servery mého ISP
```

Následující zónu vytvoříme na konci souboru `/etc/named.conf`.

```
zone "home" in {
      file "zones/home"; #název souboru pro danou zónu
      type master;
};
```

Dalším krokem je vytvoření zónového souboru `home`. Tento soubor musíme vytvořit v adresáři `/var/lib/named/zones/`. Také musíme provést změnu vlastníka a skupiny na uživatele a skupinu **named**.

Obsah souboru `/var/lib/named/zones/home` bude následující:

```
$TTL 86400        #doba platnosti záznamu
$ORIGIN .
home            IN SOA server.home.   root.server.home.(
                                2009041200      ; serial
                                10800           ; refresh
                                3600            ; retry
                                604800          ; expiry
                                86400 )         ; minimum
home.           IN NS           server.home.
home.           IN MX     1     server.home.
$ORIGIN home.
server          IN A            192.168.0.2
dana            IN A            192.168.0.5
jirka           IN A            192.168.0.6
```

Tímto jsme provedli nastavení DNS serveru. Automatický start DNS serveru nastavíme stejným způsobem jako u DHCP serveru.

7.4. Nastavení souborového serveru Samba

Pro používání samba serveru musíme specifikovat doménu, pro kterou bude tento server fungovat, a nastavit sdílené složky. Při nastavování musíme brát ohled na bezpečnost souborů. Veškeré nastavení samba serveru je v souboru `/etc/samba/smb.conf`. V globálním nastavení nastavíme:

```
workgroup = home              #nastavení domény
null passwords = no           #zakáže přihlášení prázdným heslem
map to guest = never          #zakazuje připojení pod guest účtem
dos charset = CP852           #nastavení znakové sady pro dos
```

Na konec souboru můžeme začít přidávat sdílené složky. To lze provést například takto:

```
[dana_prace]
      path = /home/user/prace
```

```
valid users = dana      #seznam oprávněných uživatelů
writable = yes          #oprávnění zapisovat
browseable = yes        #složku uvidí i ostatní v síti
create mask = 0700
#maska specifikuje práva pro nové soubory, narozdíl od masky
v Linuxu nastavuje přístupová práva
```

Po této základní konfiguraci musíme vytvořit uživatelům uživatelské účty a hesla. Vytvoření a aktivování uživatele dana provedeme následujícími příkazy:

```
smbpasswd –a dana            #následuje vytvoření hesla
smbpasswd –e dana            #aktivace účtu
```

Nyní musíme nastavit aktivování Samba serveru při startu operačního systému. To provedeme stejně jako u serveru DHCP.

7.5. Nastavení webového serveru

Nastavení webového serveru se skládá z instalace a konfigurace modulů pro použité webové aplikace. Já jsem zvolil **php5** modul a **userDir** modul. **Php5** modul umožňuje provozovat php webové aplikace. **UserDir** modul umožňuje prezentaci webových stránek jednotlivých uživatelů.

Po instalaci modulu je nutné zajistit automatické natažení tohoto modulu při spuštění web serveru. To provedeme přidáním následujících řádků do souboru `/etc/apache2/sysconfig.d/loadmodule.conf`.

```
LoadModule userdir_module   /usr/lib/apache2-prefork/mod_userdir.so
LoadModule php5_module      /usr/lib/apache2/mod_php5.so
```

Pro využití modulu **userDir** musíme provést nastavení v souboru `/etc/apache2/mod_userdir.conf`. Zde musíme do těla skriptu `<IfModule mod_userdir.c>` dopsat následující řádek:

```
UserDir enabled jirka jiri dana sarka
```

Pro zobrazení dalších webových stránek je nutné provést mapování fyzického adresáře s webovými stránkami do virtuálního adresáře web serveru. Například chceme umožnit přístup k webové aplikaci, která je umístěna `/srv/www/htdocs/squirrelmail/` z webového adresáře `/posta`.

```
Alias /posta /srv/www/htdocs/squirrelmail/
<Directory /srv/www/htdocs/squirrelmail/>
      Options Indexes
```

```
AllowOverride None
DirectoryIndex index.php
Order Allow,Deny
Allow from all
```

Po dokončení nastavení ukončíme a znovu nastartujeme HTTP server pomocí příkazu

```
service apache2 restart
```

Nyní musíme nastavit automatický start. To provedeme stejně jako v případě DHCP serveru.

7.6. Nastavení poštovního serveru

Pro konfiguraci poštovního serveru je nutné znát jméno domény, kterou bude poštovní server obsluhovat (v našem případě malakovi.cz) a musí existovat MX záznam pro tuto doménu v DNS (internet). Tento poštovní server bude umožňovat pouze lokální stahování pošty. Z tohoto důvodu nepoužíváme zvýšené zabezpečení.

Základní konfiguraci poštovního serveru provedeme v souboru `/etc/postfix/main.cf` přidáním následujících řádků

```
mydomain = malakovi.cz        #obsluhovaná doména
myorigin = $mydomain          #co se má ukazovat za @
mydestination = $myhostname, localhost.$mydomain, localhost,
      $mydomain               #pro co všechno server zpracuje poštu
inet_interfaces = all         #nastavení síťových rozhraní
mynetworks = 192.168.1.0/24, 127.0.0.0/8
home_mailbox = posta/
```

Nyní můžeme zkusit spustit postfix a otestovat, jestli jsme nastavení provedli správně. Postfix spustíme příkazem

```
postfix start
```

Zkoušku provedeme posláním zkušební zprávy pomocí příkazu **sendmail**. Pokud zpráva dorazila, je poštovní server funkční.

Dále chceme zpřístupnit poštu uživatelům přes POP3 a IMAP protokol. Pro tento účel je použit program **Dovecot**.

Hlavní nastavení programu **Dovecot** zahrnuje nastavení přístupových protokolů, cesty k poštovním schránkám a způsob autentikace uživatelů. Toto

nastavení provedeme v souboru `/etc/dovecot/dovecot.conf` přidáním následujících řádků.

```
protocols = imap imaps pop3 pop3s
ssl_disable = yes       #nebudeme používat SSL
disable_plaintext_auth = no
mail_location = maildir:/home/%n /posta/ #umístění schránek
#následující 2 položky nastavují kompatibilitu s Outlook
imap_client_workarounds = outlook-idle
pop3_client_workarounds = outlook-no-nuls oe-ns-eoh
#format UID zprávy používaný pop3
pop3_uidl_format = %08Xu%08Xv
auth default {
mechanisms = plain login     #způsob ověřování
passdb passwd-file{
#kde jsou autentikační informace pro uživatele
      args = /etc/dovecot/passwd
      }
userdb static {   #konfigurace a účet pro přístup k mailboxům
      args = uid=vmail gid=vmail home=/home/%n/posta
      }
```

Vytvoříme soubor `/etc/dovecot/passwd`, který obsahuje uživatele a hesla. V mém případě vypadá následovně:

```
jirka:{PLAIN}heslojirka
jiri:{PLAIN}heslojiri
sarka:{PLAIN}heslosarka
dana:{PLAIN}heslodana
```

Nyní máme program **dovecot** připravený ke spuštění. Spuštění provedeme pomocí:

```
service dovecot start
```

Následně provedeme přidání odkazu na startovací skript do adresáře `/etc/rc.d/rc5.d` a do adresáře `/rc3.d`.

8. Závěr

Cílem této práce bylo popsat správu operačního systému Linux a prakticky ji využít pro konfiguraci domácího serveru.

Autor popsal základní administraci operačního systému včetně nezbytného popisu příslušné problematiky a použitých příkazů. Pro praktické ověření byla použita distribuce openSUSE 10.3. Většina příkazů je nezávislá na použité distribuci Linuxu.

Práce se zabývá i administrací základních síťových služeb. Do této administrace nebyla zahrnuta administrace firewallu, který je v malých sítích obvykle realizován na přístupovém routeru/modemu k internetu. Pro zajímavost i tato jednoúčelová komunikační zařízení často používají operační systém Linux.

V praktické části je popsána konfigurace "domácího" serveru, který zajišťuje veškeré potřebné služby pro malou síť, včetně služeb potřebných pro využívání internetu (poštovní server, WEB server). Tato konfigurace plně splňuje požadavky, které si dal autor jako cíl. Server by bylo možné rozšířit o aplikaci pro přístup k poštovním schránkám přes webové rozhraní (například SquirrelMail). Dále by bylo vhodné použít i opatření proti virům a opatření proti spamu (nevyžádané poště). Vzhledem k omezenému rozsahu práce toto není součástí konfigurace serveru.

9. Seznam literatury

[1] NEMETH, E. – SNYDER, G. – HEIN, T. *Kompletní příručka administrátora.* 2. vyd. Brno: Computer Press a.s., 2008. 984 s. ISBN 978-80-251-2410-9.

[2] ČERMÁK, M. *Správa uživatelských kont. Linuxové noviny 11/2000* [online]. 2001. [cit. 30.3.2009]. <http://www.linux.cz/noviny/2000-11/clanek07.html>.

[3] FRAMPTON, S. *Linux Administration Made Easy: 6.6. Linux Password & Shadow File Formats* [online]. 2001. [cit. 30.3.2009]. <http://tldp.org/LDP/lame/LAME/linux-admin-made-easy/shadow-file-formats.html>.

[4] KYSELA, M. *Sedláme Linux, 5. díl: uživatelé a skupiny* [online]. 2003. [cit. 30.3.2009]. <http://www.zive.cz/Titulni-strana/Sedlame-Linux-5-dil-uzivatele-a-skupiny/sc-21-a-112712>.

[5] SHAH, S. – SOYINKA, W. Administrace systému Linux. Praha: Grada Publishing, 2007. 428 s. ISBN 978-80-247-1694-7.

[6] VESELÝ, A. Operační systémy II. Praha: ČZU-PEF, 2006. 255 s. ISBN 978-80-213-1553-2.

[7] JELÍNEK, L. *Jak správně rozdělit disk* [online]. 2007. [cit. 2.4.2009]. <http://www.linuxexpres.cz/praxe/jak-spravne-rozdelit-disk>.

[8] BERNÁTHOVÁ, A. *Linuxové souborové systémy* [online]. 2006. [cit. 3.4.2009]. <http://www.linuxexpres.cz/praxe/linuxove-souborove-systemy>.

[9] MythTV. *File storage* [online]. 2009. [cit. 4.4.2009]. <http://www.mythtv.org/wiki/File_storage#File_systems>.

[10] JURÁŠEK, P. *RPC, NFS a automount* [online]. 2002. [cit. 4.4.2009]. <http://www.fi.muni.cz/~kas/p090/referaty/2002-jaro/nfs.html>.

[11] Kolektiv autorů. *Příručka správce sítě: 11. Síťový souborový systém (NFS)* [online]. 1999. [cit. 4.4.2009]. <http://www.nuc.elf.stuba.sk/lit/ldp/03/030-11.htm>.

[12] PALÁT, P. *ACL v Linuxu* [online]. 2003. [cit. 5.4.2009]. <http://www.linuxzone.cz/index.phtml?idc=839&ids=29&noincread=1&showcomments=1&allcomm=1#komentare>.

[13] Zebra system s.r.o. *Inkrementální – přírůstková záloha* [online]. 2009. [cit. 7.4.2009]. <http://www.acronis.cz/kb/inkrementalni-zaloha/>.

[14] MLEJNEK, M. *Zálohování dat* [online]. 2007. [cit. 7.4.2009]. <http://www.swmag.cz/150/zalohovani-dat>.

[15] PETŘÍČEK, M. Když jeden disk nestačí [online]. 2001. [cit. 7.4.2009]. <http://www.root.cz/clanky/kdyz-jeden-disk-nestaci/>.

[16] PRŮDEK, M. *Binární balíčky (tutoriály na Rootu)* [online]. 2009. [cit. 10.4.2009]. <http://tutorialy.root.cz/linux-na-serveru/binarni-balicky/>.

[17] NEČAS, D. *Rukověť baliče RPM - I (Úvod)* [online]. 2005. [cit 10.4.2009]. <http://www.abclinuxu.cz/clanky/navody/rukovet-balice-rpm-i-uvod>.

[18] HRONČOK, M. *Vše o Hardware - Linux a správa softwaru* [online]. 2008. [cit. 10.4.2009]. <http://vseohw.net/clanky/software/linux-a-sprava-software>.

[19] *Instalace aplikací v prostředí Linuxu* [online]. 2006. [cit. 10.4.2009]. <http://www.owebu.cz/linux/vypis.php?clanek=901>.

[20] OTT, V. *Na co se často ptáme: Balíčkovací systémy* [online]. 2005. [cit 10.4.2009]. <http://www.abclinuxu.cz/clanky/navody/na-co-se-casto-ptame-balickovaci-systemy>.

[21] KOLAJA, M. *Balíčkovací systém distribuce Debian GNU/Linux: volby "source" a "build-dep" – Root.cz* [online]. 2003. [cit. 10.4.2009]. <http://www.root.cz/clanky/balickovaci-system-debian-gnu-linux-source-build-dep/>.

[22] HÄRING, D. *LINUXZONE – Úvod do systému DNS a konfigurace resolveru (1.část)* [online]. 2002. [cit 14.4.2009]. <http://www.linuxzone.cz/index.phtml?ids=9&idc=427>.

[23] HÄRING, D. *LINUXZONE – Úvod do systému DNS a konfigurace resolveru (2.část)* [online]. 2003. [cit 14.4.2009]. < http://www.linuxzone.cz/index.phtml?idc=522&ids=9>.

[24] NOVOSAD, P. *Nastavení DNS* [online]. 2003. [cit. 14.4.2009]. <http://www.abclinuxu.cz/clanky/site/nastaveni-dns>.

[25] HUBENÝ, J. *DNS* [online]. 2002. [cit 14.4.2009]. <http://www.fi.muni.cz/~kas/p090/referaty/2002-jaro/dns.html>.

[26] BANÍK, D. *DNS – 3.část* [online]. 2006. [cit. 14.4.2009]. <http://www.owebu.cz/linux/vypis.php?clanek=888>.

[27] PETŘÍČEK, M. *Lehký úvod do elektronické pošty – Root.cz* [online]. 2001. [cit. 15.4.2009]. <http://www.root.cz/clanky/lehky-uvod-do-elektronicke-posty>.

[28] Kolektiv autorů. *Postfix Architecture Overview* [online]. 2009. [cit. 15.4.2009]. <http://www.postfix.org/OVERVIEW.html>.

[29] TERPSTRA, J. *Introduction* [online]. 2003. [cit. 16.4.2009]. <http://us3.samba.org/samba/docs/man/Samba-HOWTO-Collection/IntroSMB.html#id2543643>.

10. Přílohy

10.1. Příloha A – soubor /etc/passwd

```
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
avahi:x:103:106:User for Avahi:/var/run/avahi-daemon:/bin/false
beagleindex:x:106:108:User for Beagle indexing:/var/cache/beagle:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
dhcpd:x:102:65534:DHCP server daemon:/var/lib/dhcp:/bin/false
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
games:x:12:100:Games account:/var/games:/bin/bash
haldaemon:x:101:102:User for haldaemon:/var/run/hal:/bin/false
ldap:x:76:70:User for OpenLDAP:/var/lib/ldap:/bin/bash
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
messagebus:x:100:101:User for D-Bus:/var/run/dbus:/bin/false
news:x:9:13:News system:/etc/news:/bin/bash
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash
ntp:x:74:105:NTP daemon:/var/lib/ntp:/bin/false
polkituser:x:105:107:PolicyKit:/var/run/PolicyKit:/bin/false
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
root:x:0:0:root:/root:/bin/bash
sshd:x:71:65:SSH daemon:/var/lib/sshd:/bin/false
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
vscan:x:65:103:Vscan account:/var/spool/amavis:/bin/false
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
jiri:x:1002:100:jiri:/home/jiri:/bin/bash
perforce:x:1006:2:perforce:/home/perforce:/bin/bash
vmail:x:5000:5000:vmail:/home/vmail:/bin/bash
```

10.2. Příloha B – soubor /etc/shadow

```
at:!:13908:0:99999:7:::
beagleindex:!:13908:0:99999:7:::
bin:*:13908:::::
daemon:*:13908:::::
dhcpd:!:13908:0:99999:7:::
ftp:*:13908:::::
lp:*:13908:::::
mail:*:13908:::::
man:*:13908:::::
mysql:!:13908:0:99999:7:::
named:!:13908:0:99999:7:::
news:*:13908:::::
```

```
nobody:*:13908::::::
ntp:!:13908:0:99999:7:::
polkituser:!:13908:0:99999:7:::
postfix:!:13908:0:99999:7:::
root:$2a$05$jGqouredj5vaL6BjlntTiN8HHcu:13908::::::
suse-ncc:!:13908:0:99999:7:::
uucp:*:13908::::::
vscan:!:13908:0:99999:7:::
wwwrun:*:13908::::::
jiri:$2a$05$TYo3l0Kouhoh58HTeM3bh:13932:0:99999:7:::
perforce:$2a$05$C2/:0:99999:7:::
test:$2a$05$dYYHUf98vT/woOCFgiPly5pnnC:13908:0:99999:7:::
vmail:$2a$05$/ xeMcDVx3g5Fzga:13915:0:99999:7:::
```

10.3. Příloha C – soubor /etc/fstab

```
/dev/sda1      /                    ext3          acl,user_xattr        1 1
/dev/sda2      /home                ext3          acl,user_xattr        1 2
/dev/sda3      swap                 swap          defaults              0 0
proc           /proc                proc          defaults              0 0
sysfs          /sys                 sysfs         noauto                0 0
debugfs        /sys/kernel/debugde  bugfs         noauto                0 0
usbfs          /proc/bus/usb        usbfs         noauto                0 0
devpts         /dev/pts             devpts        mode=0620,gid=5       0 0
```

10.4. Příloha D – příklad instalace balíčků

```
Linux:~ # rpm -ivh /home/mali/RPMs/pin-0.38-29.noarch.rpm
Preparing...                ########################################### [100%]
   1:pin                    ########################################### [100%]
Linux:~ # zypper in dovecot
* Reading repository 'Packman Repository' cache
* Reading repository 'KDE4' cache
* Reading repository 'Main Repository (OSS)' cache
* Reading repository 'openSUSE-10.3-Eval 10.3' cache
* Reading repository 'Main Repository (NON-OSS)' cache
* Reading repository 'VideoLan Repository' cache
* Reading repository 'KDE4com' cache
* Reading repository 'openSUSE-10.3-Updates' cache
* Reading repository 'openSUSE BuildService - KDE:Community' cache
* Reading repository 'openSUSE BuildService - KDE:Backports' cache
* Reading repository 'KDE4extra' cache
* Reading installed packages [100%]


The following NEW packages are going to be installed:
  dovecot postgresql-libs

Overall download size: 1.9 M. After the operation, additional 4.5 M will be used.
Continue? [yes/no]: yes
Downloading package postgresql-libs-8.2.13-0.1.i586, 190.3 K (495.0 K unpacked)
Downloading: postgresql-libs-8.2.13-0.1.i586.rpm
* Downloading [100%]
* Installing: postgresql-libs-8.2.13-0.1 [100%]
Downloading package dovecot-1.0.5-6.4.i586, 1.7 M (4.0 M unpacked)
Downloading: dovecot-1.0.5-6.4.i586.rpm
* Downloading [100%]
* Installing: dovecot-1.0.5-6.4 [100%]
```