Technická univerzita v Liberci

FAKULTA PEDAGOGICKÁ

Katedra: Aplikované informatiky
Studijní program: 2. stupeň ZŠ
Kombinace: Anglický jazyk + Informatika

Návrh malé počítačové sítě a popis její realizace na ZŠ

Design and Description of a Small Computer Network and its Implementation at Basic School

Autor:
Kamil Kantar

Podpis:

Adresa:
Na Lučinách 653
417 12 Proboštov

Vedoucí práce: Mgr. David Kmoch
Konzultant: Ing. Petr Kretschmer

Počet

Stran	Slov	Obrázků	Tabulek	Pramenů	Příloh
72	22789	6	5	13	7

V Liberci dne: 4.5.2006

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci dne: .....

Poděkování

Touto cestou bych chtěl poděkovat vedoucímu diplomové práce Mgr. Davidu Kmochovi a konzultantovi Ing. Petru Kretschmerovi za veškerou pomoc a odborné připomínky při její tvorbě.

Dále bych chtěl poděkovat Mgr. Martinu Lánovi ze Základní školy Máchovo náměstí Děčín za ochotu při poskytování všech informací nutných k vypracování této práce.

Mé poděkování si zaslouží rovněž D. E. Knuth za vynikající typografický systém TEX, Bram Moolenaar za textový editor `VIM` a Patrick Volkerding za vývoj linuxové distribuce Slackware.

NÁVRH A POPIS MALÉ POČÍTAČOVÉ SÍTĚ A POPIS JEJÍ REALIZACE NA ZŠ

KANTAR Kamil DP-2006 **Vedoucí DP**: Mgr. David Kmoch

Resumé

Diplomová práce se zabývá návrhem a popisem síťového řešení pro učebnu výpočetní techniky na ZŠ Máchovo náměstí v Děčíně. Síť bude zároveň implementována do stávajícího prostředí ve škole. Realizace této počítačové sítě se řídí požadavky školy a cílem návrhu je takové řešení, které by vyhovovalo podmínkám pro výuku výpočetní techniky ve zmíněné základní škole. Základním článkem popisované sítě je souborový a aplikační server s operačním systémem Linux, který bude své služby nabízet 15 stanicím v počítačové učebně výpočetní techniky. Popis instalace tohoto serveru tvoří hlavní část práce. Zbytek líčí konfiguraci pracovních stanic, na kterých je možno využívat operační systémy MS Windows a Linux.

DESIGN AND DESCRIPTION OF A SMALL COMPUTER NETWORK AND ITS IMPLEMENTATION AT BASIC SCHOOL

Summary

This Diploma Thesis deals with design and description of a computer network solution for IT classroom at basic school Máchovo náměstí Děčín. This solution will be implemented in the existing computer network layout in the school. This computer network implementation follows school's requirements and the goal of the design is a solution that would correspond with conditions for teaching IT at the school. The basic element of the described network is a file and aplication server with Linux operating system. It offers its services to 15 desktop computers in the classroom. This computer's installation description makes the main part of this diploma thesis. The rest of it deals with desktop computers configuration, on which both MS Windows and Linux operating systems will be available.

Obsah

Seznam tabulek

Seznam obrázků

1 Úvod

Když jsem před několika lety poprvé pracoval s operačním systémem Linux a jeho síťovými prostředky a možnostmi, nepomyslel jsem si, že by mě tyto činnosti zaujaly natolik, abych byl schopný a ochotný psát diplomovou práci zaměřenou tímto směrem. Problematika mě však postupem času zaujala natolik, že jsem se pro téma počítačové sítě (s Linuxem jako svým základním článkem) nakonec rozhodl.

V průběhu své souvislé pedagogické praxe jsem působil na Základní škole Máchovo náměstí v Děčíně, kde jsem učil anglický jazyk a informatiku. Jako v jedné z mála základních škol se na této vyučuje vedle operačního systému Microsoft Windows také Linux. Tamější vedoucí úseku výpočetní techniky a školní počítačové sítě Mgr. Martin Lána byl mým návrhem vypracování diplomové práce s ohledem na současnou situaci a potřeby této školy mile překvapen. Poté, co jsem mu stručně popsal strukturu a vlastnosti svého návrhu, rozhodl se od příštího akademického roku (tedy 2006/2007) toto řešení ve škole implementovat. Současný stav jednoho úseku počítačové sítě týkající se učebny výpočetní techniky totiž z několika důvodů nevyhovuje jeho představám pro výuku tohoto předmětu ve škole.

Cílem této diplomové práce je navrhnout a podrobně popsat úsek počítačové sítě, která bude ve všech ohledech bezpečná a zároveň vyhovující potřebám výuky ve zmíněné základní škole. Největší důraz jsem kladl na požadavky školy a do sítě jsem implementoval veškeré prvky, nutné pro bezproblémovou výuku v počítačové učebně. Samozřejmostí je bezpečnost sítě a její odolnost proti útokům vnějším i vnitřním.

2 Současný stav

Ve škole jsou dvě počítačové učebny, ve kterých probíhá veškerá výuka předmětu *Výpočetní technika* a odpoledního zájmového kroužku s názvem *Informatika*. Podle pana Lány i vedení školy je počet učeben dostačující a jejich kapacity, ve smyslu počtu pracovních stanic, jsou vyhovující. První učebna – s názvem VT1 – je vybudována v rámci projektu INDOŠ, zatímco učebna druhá – VT2 – je vybavená vlastními školními počítači.

2.1 Učebna VT1

Vybavení v rámci projektu INDOŠ je v učebně využíváno od roku 2002. „Internet do škol“ je projekt Ministerstva školství, mládeže a tělovýchovy, který prostřednictvím generálního dodavatele (nyní provozovatele) zavedl do počítačově nevybavených českých škol počítače, periferie, internetové připojení a související internetové i intranetové služby [8]. Učebna je vybavena 10 žákovskými počítači, 1 učitelským, síťovou tiskárnou, serverem a je připojena na internet. Jako operační systém je zde nainstalován Microsoft Windows 2000. Slouží především k výuce žáků prvního stupně a využívá pouze omezené programové vybavení dodávaného v rámci projektu INDOŠ. Seznam softwaru, registrovaného pro instalaci a provoz ve školské síti INDOŠ, je k dispozici na `http://www.indos.cz/katalogsw/`. Původní plán mé diplomové práce zahrnoval učebny obě, avšak učebnou VT1 se zabývat nebudu, protože:

- Smlouva s MŠMT zakazuje provádět jakékoliv změny v učebnách INDOŠ před jejím ukončením.
- Podle pana Lány současné provedení a funkčnost učebny VT1 výuce víceméně vyhovuje a není třeba jej v současné době měnit.

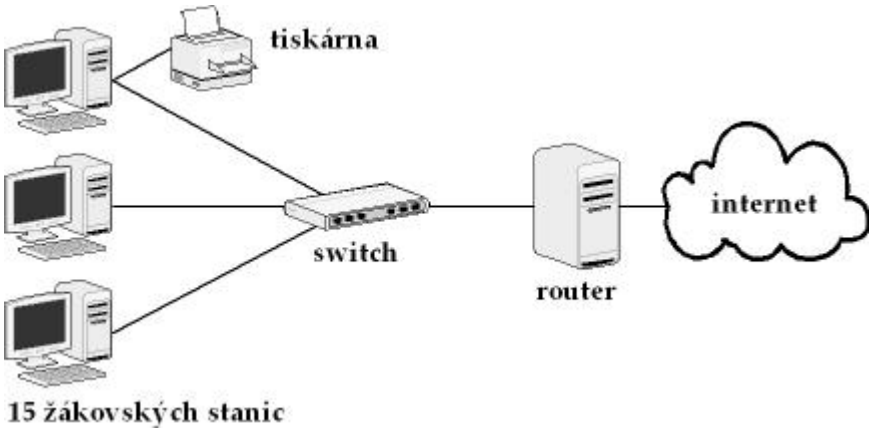
2.2 Učebna VT2

Realizace nové počítačové sítě se týká učebny VT2. Ta nyní obsahuje 15 žákovských stanic, tiskárnu a síťový směrovač. V současné době je na všech klientských stanicích nainstalován operační systém SuSE Linux 9.3, který zajišťuje i chod směrovače. Propojení všech počítačů v učebně je realizováno pomocí Ethernetu na kroucených dvoulinkách (100 Mb/s) a síťového switche Eidmax s 24 porty RJ-45. Směrovač má v rámci internetu přidělenu veřejnou IP adresu, a tak zajišťuje konektivitu vnitřní sítě pomocí SNATu. Schéma současného stavu učebny je znázorněno na obrázku č. 1.

2.2.1 Nevýhody současného stavu v učebně VT2

V úvodu jsem zmínil, že současný stav v učebně VT2 nevyhovuje požadavkům školy. Jedná se o tyto nedostatky:

- Přítomnost pouze jednoho operačního systému na žákovských stanicích (Linux). S tím samozřejmě souvisí nemožnost využití výukových programů, které chce škola



Obrázek 1: Současný stav v učebně VT2

využívat v mnoha předmětech (cizí jazyky, zeměpis, přírodopis, ...) v čase, kdy je obsazena učebna VT1. Většina takovýchto výukových programů je totiž určena pouze pro systém Microsoft Windows. Zároveň však není možné ponechat na počítačích v učebně pouze systém Windows, protože jedním z hlavních cílů tříd s rozšířenou výukou výpočetní techniky je práce s operačním systémem Linux a programech pro něj určených – tedy rozšíření běžného učiva základních škol.

- Nevyhovující situace s uživatelskými účty žáků. Uživatelské účty jsou pouze lokální a proto je nutné, aby žáci při každé hodině seděli vždy u stejného počítače.
- S tím je spojena další nevýhoda – nemožnost sdílení domovských adresářů uživatelů mezi jednotlivými počítači. Sdílení souborů probíhá pouze pomocí jednoho sdíleného adresáře ze směrovače, který se přes systém NFS připojuje ke všem stanicím.
- Nízká bezpečnost sítě – není nastaven firewall ani žádné jiné zabezpečení vnitřní sítě.
- Není zajištěna „filtrace“ internetového obsahu – žáci tedy mohou přistupovat na libovolné internetové stránky (tedy i na ty s nevhodným obsahem).
- Tiskárna je připojena pouze k jedné stanici a není sdílená v síti. Tisk z ostatních stanic je tedy nutné provést zkopírováním požadovaného souboru na sdílený disk a vytisknout na počítači s tiskárnou.

- Decentralizovaná správa sítě. Podle pana Lány by bylo výhodné většinu vlastností nové sítě spravovat z jednoho místa a tím celkovou administraci zpřehlednit a ulehčit. Jedná se o běžné činnosti jako je přidávání (odebírání) uživatelů, změna jejich hesel, změna uživatelských práv pro síťové prostředky, přidělování síťových informací stanicím ze serveru apod.

Navrhované síťové řešení odstraňuje všechny tyto problémy a přidává navíc několik vlastností, které usnadní správu sítě a zvýší její celkovou bezpečnost.

2.3 Ostatní místa s počítači

Je nutné zmínit, že škola samozřejmě disponuje několika dalšími počítači, které jsou umístěny zejména v učitelských kabinetech. Bohužel podle Mgr. Lány vedení školy neplánuje propojení těchto počítačů s učebnou VT2, a tudíž se přímo netýkají návrhu části počítačové sítě popisované v této práci.

3 Navrhované řešení pro učebnu VT2

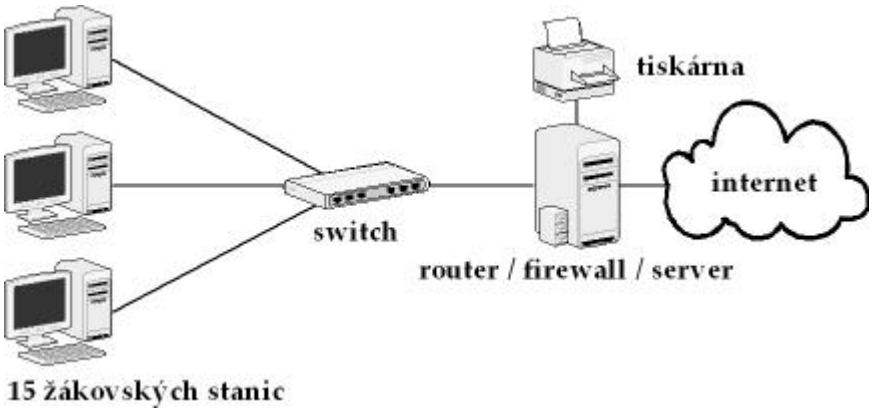
Na obrázku č. 2 je znázorněno poupravené schéma současné počítačové sítě v učebně VT2. To umožní síť zrealizovat tak, aby bylo možné odstranit všechny zmíněné související nedostatky. Směrovač bude nyní také plnit funkci souborového a aplikačního serveru a poskytovat tyto služby všem žákovským počítačům v učebně. Bude k němu připojena tiskárna, sdílená pro všechny klienty. Bezpečnost vnitřní sítě bude zajištěna firewallem a dalšími zabezpečovacími prvky, které budou součástí serveru.

3.1 Služby serveru

Server v učebně VT2 bude poskytovat následující služby:

- Přidělování síťových identifikačních informací žákovským počítačům. Díky službě DHCP (Dynamic Host Configuration Protocol) budou všechny nezbytné informace včetně IP adres přiřazovány dynamicky ze serveru.

- Funkce primárního doménového řadiče domény NT pro klientské stanice s operačním systémem Microsoft Windows XP.



Obrázek 2: Navrhované schéma pro učebnu VT2

- Služby souborového a tiskového serveru.
- Centralizovaná správa uživatelských účtů pro klienty s operačním systémem Linux včetně exportu domovských adresářů.
- Synchronizace uživatelských účtů pro oba operační systémy.
- Směrovač / firewall pro bezpečné připojení sítě k internetu.
- Spravování síťové komunikace (http požadavků) pomocí Proxy serveru.

Toto jsou hlavní vlastnosti a služby nového serveru v učebně VT2. Zde je nutné zmínit, že na všech žákovských počítačích bude možno zavést jakýkoliv ze dvou operačních systémů Microsoft Windows nebo Linux (jedná se o tzv. *dual boot*).

4 Proč právě Linux?

Hlavní počítač v učebně VT2 sloužící jako server / firewall / směrovač (čili *router*) se bude v mé práci pro jednoduchost a přehlednost jmenovat `orpheus` (samozřejmě se jedná o stejný stroj, který dosud sloužil pouze jako směrovač). Jak již bylo řečeno, `orpheus` bude postaven na operačním systému Linux, který umožní relativně jednoduchou implementaci všech zmíněných prostředků a služeb.

4.1 Vztah Linuxu k Unixu

O operačním systému Linux se dá říci, že je to verze Unixu vytvořená jako implementace normy POSIX fungující na různých hardwarových platformách, a je slučitelná s většinou existujícího softwaru pro Unix. Od většiny ostatních verzí Unixu se liší v tom, že je svobodná (ve smyslu svobodného software), open source, a že je vyvíjena kooperativně, přičemž příspěvky přicházejí od stovek jednotlivců a organizací.

Především díky projektu GNU byla většina důležitého softwaru dodávajícího unixovým systémům velkou užitnou hodnotu vyvinuta v některé z podob modelu otevřeného softwaru (open source). Tentýž zdrojový kód funguje na linuxových i nelinuxových systémech. Například webový server Apache se vůbec nestará o to, zda běží na Linuxu nebo na systému HP-UX. Z hlediska proprietárních aplikací je Linux prostě jedna z nejlépe podporovaných variant Unixu.

[7]

4.2 Historie Linuxu a Unixu

Linux vznikl v roce 1991 jako osobní projekt Linuse Torvaldse, finského vysokoškolského studenta. Linus Torvalds původně tento projekt plánoval jako odnož Minixu, což byl modelový operační systém, jehož autorem byl Andrew S. Tanenbaum. Linux ale vyvolal v celém světě velký zájem a toto jádro začalo brzy žít svým vlastním životem. Linus využil možností kooperativního vývoje, a proto mohl dosáhnout mnohem ambicióznějších cílů. V roce 1994 byla vydána verze jádra 1.0; v době psaní této práce (počátek roku 2006) má nejnovější stabilní verze linuxového jádra označení 2.6.15.

Linux zdědil mnoho funkcí od svých unixových předků, a proto není zcela spravedlivé umísťovat počátek linuxové éry do roku 1991. Historie Unixu sahá několik desetiletí do minulosti, konkrétně do roku 1969, kdy UNIX vznikl jako výzkumný projekt v AT&T Bell Labs. V roce 1976 byl UNIX zdarma poskytnut univerzitám a stal se tak základem pro mnoho kurzů o operačních systémech a akademických výzkumných projektů.

[7]

4.3 Linuxové distribuce

Jedním z hlavních rozdílů mezi Linuxem a ostatními variantami Unixu je, že slovem *Linux* se vlastně popisuje jen samotné jádro operačního systému. Avšak k plnohodnotnému

Distribuce	Web	Komentář
Debian GNU/Linux	`http://www.debian.org`	Distribuce orientovaná na stabilitu a firemní nasazení. Kolem Debianu existuje obrovská komunita.
Mandriva Linux	`http://www.mandriva.com`	Orientace na desktopové prostředí, vazba na grafickou nástavbu.
Fedora Linux	`http://www.redhat.com`	Pokračovatel kdysi nejrozšířenější distribuce RedHat.
Slackware Linux	`http://www.slackware.com`	Nejstarší distribuce, stabilní, pro pokročilé uživatele.
SuSE Linux	`http://www.suse.com`	Oblíbená především v Evropě, spousta grafických ovládacích prvků.

Tabulka 1: Nejoblíbenější, všeobecně zaměřené distribuce

a použitelnému operačnímu systému se dostaneme až tehdy, když k tomuto jádru přidáme nějaký software. Takovému celku se pak říká *distribuce*. Je tedy pochopitelné, že všechny linuxové distribuce sdílejí stejný jaderný rodokmen, ale zbytek tvoří nejrůznější doplňující materiál, který se v rámci jednotlivých distribucí může velice lišit. Ty tak mohou dostávat odlišné rozměry z hlediska zaměření, filosofie, použitelnosti či podpory a obliby. Tabulka č. 1 obsahuje seznam dlouhodobě nejoblíbenějších, všeobecně zaměřených distribucí.

V tabulce samozřejmě zdaleka nejsou uvedeny mnohé další oblíbené distribuce. Důkladný průvodce nejrůznějšími linuxovými distribucemi lze vždy nalézt na internetových adresách `http://linux.com/dist` nebo `http://www.distrowatch.com`.

4.4 Linux na serveru

Linux je perfektní volbou operačního systému, na kterém se dá vybudovat síťový sever, protože poskytuje téměř všechny důležité síťové služby v jediném levném balíku[1]. Nízká

[1] A ty, které implicitně neobsahuje, lze velmi jednoduše doinstalovat.

cena, spolehlivost a síla pohánějí neustálý rozvoj Linuxu jako serverového systému. Linux se ukázal být efektivní alternativou drahých unixových serverů. Také se prokázalo, že je výkonnější a spolehlivější než jakýkoliv prodávaný operační systém pro pracovní stanice přetvořený na serverový operační systém. Škála síťových služeb nabízených Linuxem značí, že tento systém je možné použít k uspokojení jakýchkoliv potřeb síťového serveru.

4.4.1 Důvody nasazení Linuxu na server `orpheus`

- Linux nabízí všechny potřebné programové prostředky, aby bylo možné provozovat veškeré služby popisované v kapitole 3.1.
- Samotný operační systém i potřebný software provozovaný na počítači `orpheus` je šířen pod licencí GPL GNU, což znamená, že je zcela zdarma. Vlastní distribuci Linuxu, která bude na serveru instalována, stáhneme z internetu ve formě obrazu CD a téměř všechen potřebný software v něm bude již obsazen. Škola tak nebude muset investovat žádné peníze do nákupu serverového operačního systému ani dalšího potřebného softwaru.
- S tím souvisí nízké pořizovací náklady. Hardwarové nároky na provoz Linuxu se v dnešní době dají považovat za minimální, a tak škola nebude nucena pořizovat drahý značkový server. Veškeré požadavky na služby může nabízet na „obyčejném“ PC při stejné kvalitě výkonu, spolehlivosti a dostupnosti jako u komerčních řešení.
- Flexibilita – Je možno sestavit systém šitý na míru konkrétnímu použití na školním serveru a nainstalovat jen to, co je nezbytně nutné (s možností dalšího rozšíření), což opět zvyšuje výkon a bezpečnost.
- Na Linuxu není zpravidla nutné používat žádný komplikovaný zabezpečovací nebo antivirový systém. Linux je sám o sobě velmi dobře zkonstruován a odolává mnoha známým bezpečnostním komplikacím (např. virům). Aktualizace systému jsou dostupné zdarma a včas každému bez rozdílu.
- Vzdálená správa pomocí příkazové řádky, kterou je možné s výhodou použít aniž by byl správce serveru v učebně zrovna přítomen.
- Linux je schopen plnohodnotně zastoupit komerční Windows NT/2000 server a klientské stanice nemusí ani poznat, že na straně serveru běží Linux. Vysoký výkon

v síťových aplikacích (podpora TCP/IP je implementována přímo v jádře) systém přímo předurčuje pro použití na serverech.

- Kvalifikovaná podpora a dokumentace. S Linuxem pracuje na celém světě velké množství lidí, včetně odborníků, kteří jsou ochotni se podělit o svoje zkušenosti. Většinu dokumentace lze nalézt na internetu.

5 Instalace serveru orpheus

5.1 Hardware

Server orpheus se skládá z hardwarové konfigurace, kterou popisuje tabulka č. 2. Všechny hardwarové komponenty, ze kterých se orpheus skládá, jsou pro potřeby našeho serveru plně dostačující. Podle Mgr. Lány škola nebude investovat peníze do nákupu nových hardwarových součástí, pokud to není nezbytně nutné. Jelikož byl tento počítač původně kupován pro účely souborového serveru, jeho hardwarová konfigurace mnohokrát převyšuje nutné minimum pro síťový směrovač, což je funkce, kterou v současné době stroj plní.

Základní deska	Intel D1859
Procesor	Intel Pentium 4 s frekvencí 3 GHz
Paměť	2 x 256 MB ECC DDR2
Pevné disky	160 GB SATA
Síťové karty	2 x Intel 82559 10/100 Mbps Fast Ethernet Controller

Tabulka 2: Hardwarová konfigurace serveru orpheus

5.2 Volba distribuce

Na server orpheus jsem vybral linuxovou distribuci Slackware[2], což je nejstarší distribuce vůbec. Jedná se o prověřený a stabilní linuxový systém, jehož hlavními přednostmi jsou jednoduchost a přehlednost. Taková je i jeho vnitřní struktura a konfigurační soubory jsou často bohatě komentované, a tudíž snáz konfigurovatelné. Obsahuje jednoduchý instalační

[2]Domovská stránka Slackware Linuxu je na adrese http://www.slackware.com

program, vyčerpávající online dokumentaci a balíčkovací systém nabízející všechny funkce, které budou pro potřeby serveru nezbytné. Díky svým vlastnostem – stabilitě a bezpečnosti – je často zmiňován jako distribuce vhodná především na servery. Ve své základní instalaci jsou v něm obsaženy v podstatě všechny nástroje a programy nutné pro běh serveru `orpheus` a jeho služeb. Také moje zkušenosti s nasazením Slackware Linuxu na server jsou výhradně pozitivní.

ISO obraz Slackware Linuxu v jeho poslední stabilní verzi 10.2 lze stáhnout např. ze stránky `ftp://ftp.slackware.com/pub/slackware/slackware-10.2-iso/` a poté jej vypálit na CD. Pro potřeby serveru `orpheus` stačí stáhnout pouze první instalační CD, protože to druhé obsahuje pouze grafické prostředí KDE, které se samozřejmě na tento počítač instalovat nebude.

5.3 Rozdělení disku

Správné rozdělení pevného disku na oddíly je první krok, který je nutné podniknout k zabránění nechtěného zaplnění systémových částí disku. K tomu může dojít z několika důvodů:

1. Systémové logy zaplní svým obsahem celé volné místo.
2. Update softwaru nainstaluje časem více balíčků, než je schopen disk snést.
3. Uživatelé zaplní disk svými daty, ať už schválně nebo z neznalosti.[3]

Proti těmto problémům se lze bránit vhodným rozdělením disku na jednotlivé oddíly, z nichž každý bude obsahovat část dat. Toto rozdělení většinou probíhá při instalaci operačního systému a vždy je třeba mít již předem rozdělení disku naplánováno podle skutečných potřeb. Tabulka č. 3 popisuje běžné oddíly systému Linux a v tabulce 4 je uvedeno konkrétní rozdělení disku na serveru `orpheus`.

5.4 Volba souborového systému

Filesystém (souborový systém) zajišťuje uložení informací (ve formě souborů a adresářů) na disk a následně jejich zpětné vyvolání. U filesystémů, které umí Linux „nativně“ vy-

[3]Tento bod je v našem případě zvláště kritický, uvědomíme-li si, že uživateli, kterým bude `orpheus` své služby nabízet, jsou žáci základní školy.

Oddíl	Popis
`swap`	Povinný oddíl, slouží jako rozšíření paměti RAM pro odkládání neaktivních procesů.
`/`	Povinný oddíl (též nazývaný `root`) je kořenem celého souborového systému.
`/boot`	Soubory potřebné pro start systému se někdy umísťují do zvláštního oddílu.
`/usr`	Oddíl obsahuje většinu systémového softwaru.
`/home`	Obsahuje všechny domovské adresáře běžných uživatelů a také většinu uživatelských souborů.
`/var`	Soubory určené pro tisk, poštu a soubory news a systémové logovací soubory.
`/opt`	Obsahuje volitelný software. Některý dodatečný software předpokládá existenci adresáře `/opt` a instaluje se do něho.

Tabulka 3: Běžné oddíly. Zdroj: [5, s. 568]

užívat, a které připadají v úvahu pro server `orpheus`, je třeba zvážit několik důležitých aspektů:

- Stabilita (data by se rozhodně neměla ztrácet běžným používáním).
- Dostatečný výkon (z mnoha hledisek).
- Funkčnost souborových kvót.
- Výhodou je podpora *žurnálování*. Žurnálovací filesystém si uchovává informace o operacích, které provedl a je pak v případě výpadku schopen dostat se rychle zpět do konzistentního stavu.
- Podpora Slackware Linuxem, resp. verzí 10.2 (ale vzhledem k tomu, že Slackware podporuje všechny souborové systémy, které lze brát pro `orpheus` v úvahu, můžeme tento bod v klidu vypustit.)

Na všechny diskové oddíly serveru `orpheus` jsem vybral souborový sytém `ext3`.

5.4.1 Souborový systém `ext3`

Filesystém `ext3` je rozšířením standardního linuxového souborového systému (`ext2`). Jeho autorem je Dr. Stephen Tweedie. `Ext3` přidává k existujícímu kódu `ext2` zmíněné žurnálovací funkce tak, že systém `ext3` vyhrazuje část disku pro žurnálový soubor. Když se

`swap`	Odkládací oddíl bude dvojnásobně veliký jako velikost operační paměti. Tedy 1024 MB.
`/`	Velikost kořenového adresáře je 700 MB. Díky oddělení adresáře `/usr` bude tato kapacita dostačující.
`/var`	Oddíl je velký 2 GB. Na počítači nebude běžet poštovní server ani se nejedná o server news. Oddíl `/var` bude využíván zejména pro paměť cache proxy serveru `squid`. Velikost logovacích souborů bude sledována.
`/usr`	Velikost 5 GB, což by mělo pro linuxový software i software doplňkový stačit.
`/tmp`	Velikost 1 GB. Do adresáře `/tmp` může každý zapisovat a tak by pro něj měl existovat samostatný oddíl. Navíc se bude tento adresář mapovat jako síťová jednotka systému Windows a tak bude žákům opravdu snadno přístupný.
`/home`	Téměř celý zbytek disku, tedy asi 140 GB, připadne na domovské adresáře uživatelů.
nealokováno	Na disku zůstane cca 5 GB volného prostoru, který je možné později využít tam, kde by docházelo místo, či pro rozšíření služeb poskytovaných serverem.

Tabulka 4: Rozdělení disku na serveru `orpheus`

objeví požadavek na operaci se souborovým systémem, požadované úpravy se nejprve zapíší do žurnálu. Když je dokončena aktualizace žurnálu, zapíše se záznam „uložit“, který označuje konec položky. Teprve pak se modifikuje normální souborový systém. Kdyby došlo k havárii, může se žurnál použít k rekonstrukci dokonale konzistentního souborového systému. Žurnálování snižuje čas potřebný na testy konzistence souborového systému[4] asi na jednu sekundu na jeden souborový systém. Pokud nedošlo k nějaké hardwarové chybě, lze souborový systém `ext3` prakticky okamžitě otevřít a rekonstruovat.

`Ext3` je dobrý kompromis mezi stabilitou a funkcemi. Rozumně v něm fungují ACL[5], žurnál je čistě dodělaný a téměř nedegraduje výkon. Výhodou také je, že ho lze bezpečně zmenšit a zvětšit, a dokonce i zvětšit připojený filesystém. `Ext3` je velmi otestovaný a

[4] Například pomocí nástroje `fsck`.

[5] *Access Control List*, rozšíření přístupových práv UNIXu (UGO), které však používat nebudeme.

široce podporovaný filesystém a jelikož v případě počítače `orpheus` neexistují speciální požadavky, je bezpochyby dobrou volbou.

5.5 Kroky po instalaci

Po instalaci linuxového operačního systému je vždy dobré provést pár operací k obecnému zvýšení bezpečnosti. Hlavním z těchto postupů je stažení bezpečnostních aktualizací nainstalovaného softwaru. V systému Slackware Linux se toto dá velice pohodlně provést pomocí programu `swaret`, který lze stáhnout na adrese `http://swaret.sf.net`.

Další bezpečnostní úpravou by mohlo být zamezení vytváření a spouštění speciálních souborů na diskových oddílech, kam mají právo zápisu obyčejní uživatelé. Soubory s příznakem `setuid` se spouštějí s právy vlastníka a nikoliv uživatele, který je spustil. To může znamenat potenciální bezpečnostní riziko (například zneužitím chyby ve špatně napsaném programu), a proto je dobré zabránit ukládání takovýchto souborů na diskové oddíly s právy zápisu obyčejných uživatelů. Toho lze docílit uvedením klíčového slova `nosuid` v parametrech připojení daného souborového systému v souboru `/etc/fstab`[6]. Jedná se o oddíly připojující se k adresářům `/home, /var` a `/tmp`. Kromě příznaku `nosuid` je vhodné zamezit vytváření systémových zařízení (devices) v těchto oddílech přidáním další vlastnosti: `nodev`. V adresářích `/tmp` a `/var` by se navíc neměly vyskytovat ani spustitelné soubory a tak poslední vlastnost při připojování příslušných oddílů má název `noexec`.

Je důležité, aby byl čas na serveru `orpheus` synchronizován s časem na stanicích (například kvůli správné funkčnosti cestovních profilů v systému Windows). K tomu je potřeba, aby na serveru běžel stále správný čas. Toho se dá docílit pravidelnou synchronizací času s nějakým veřejným časovým serverem. Pravidelné spouštění (například přes daemon `crond`) příkazu

```
ntpdate ntp.cesnet.cz
```

zajistí vše potřebné pro správné nastavení času na serveru `orpheus`.

[6] Což je textový soubor, jehož účelem je systému popsat jednotlivé diskové svazky a vysvětlit mu, kam a jakým způsobem je má připojovat.

6 Práce se skupinami a uživateli

6.1 Příprava pro souborové kvóty

Jak již bylo zmíněno, diskový prostor pro uživatele serveru `orpheus` bude limitován souborovými kvótami. Pro jejich správný chod je nutné nastavit limity kvót pro nový účet příkazem `edquota`. Tento příkaz lze používat interaktivně, ale častěji se používá v „prototypovém“ režimu, aby modeloval kvóty nového uživatele podle kvót jiného. Dříve než budou vytvořeni skuteční uživatelé, je výhodné vytvořit právě tyto prototypové, přičemž pro každou skupinu uživatelů serveru bude vytvořen jeden prototyp. Server `orpheus` disponuje 160 GB diskovým prostorem, a tak bude přidělená uživatelská kvóta rozdělena takto:

- **studenti**: 150 MB pro každého
- **učitelé**: 200 MB pro každého
- **ostatní**: 50 MB pro každého

Pro pořádek ve skupinách bude vytvořena skupina uživatelů `prototypy`, do níž budou patřit všichni modeloví uživatelé (a nikdo jiný). K tomu však není použit příkaz pro přidávání skupin `groupadd`, ale stačí pouze v souboru `/etc/group` přepsat název skupiny `users` s číslem 100 na `prototypy`. Skupina `users` nebude totiž potřeba. Poté lze již přidat modelové uživatele:

```
useradd -c "prototyp - student" -g 100 -s /bin/false prototyp_studenti;
useradd -c "prototyp - ucitel" -g 100 -s /bin/false prototyp_ucitele;
useradd -c "prototyp - ostatni" -g 100 -s /bin/false prototyp_ostatni;
```

Jak je z těchto tří příkazů vidět, jako shell byl modelovým uživatelům vybrán neplatný `/bin/false`, tedy něco, co není zapsáno v souboru `/etc/shells`. Tím se zabezpečí, že se nikdo nebude moci přihlásit k serveru jako modelový uživatel. Skutečné souborové kvóty budou vytvořeny po přidání skupin a uživatelů.

6.2 Přidání skupin a uživatelů

Všichni uživatelé budou rozděleni do 3 skupin: *studenti, učitelé* a *ostatní*. Význam skupin `studenti` a `ucitele` je zřejmý, skupina uživatelů `ostatni` bude sloužit pro osoby se zvláštním statutem na škole, významné účty s více privilegii, speciální účty pro testovací účely a podobně.

K přidávání skupin a uživatelů slouží příkaz `groupadd`, resp. `useradd`. Vzhledem k tomu, že se v tomto případě bude přidávat přibližně 450 žáků a téměř 4 desítky učitelů, byla by registrace účtů velmi zdlouhavá. K hromadnému přidávání účtů jsem vybral skript `mass-useradd.py` napsaný v jazyce Python Aaronem Maloneyem a upravil jej pro přesné potřeby registrace uživatelů na serveru `orpheus`[7]. Tento skript tvoří přílohu A diplomové práce.

Nejprve budou přidány skupiny uživatelů:

```
groupadd studenti; groupadd ucitele; groupadd ostatni
```

Tím se přidají skupiny `studenti`, `ucitele` a `ostatni`, jejichž `GID` je 102, 103 a 104[8].

Pro přidání uživatelů – studentů skriptem `mass-useradd.py` je nutné mít vytvořený textový soubor `studenti.txt`, který má na každém svém řádku údaje o jednom uživateli v tomto formátu: `prihlasovaci_jmeno: Jmeno Prijmeni`. Přihlašovací jméno uživatele se bude skládat z prvních čtyř písmen příjmení a prvních dvou písmen křestního jména. Uživatel Kamil Kantar by tedy měl přihlašovací jméno `kantka`. Pokud se bude nějaké jméno vícekrát opakovat, je nutné na jeho konec přidat číslo 1, poté 2 atd. Upravený skript `mass-useradd.py` vytvoří uživatele, nastaví mu náhodně vygenerované heslo, vytvoří mu domovský adresář v oddílu `/home` a nastaví k němu práva `700`, tedy úplné řízení pro uživatele, žádné pro ostatní. Ještě před samotným spuštěním skriptu je výhodné umístit do adresáře `/etc/skel` všechny soubory, které budeme chtít mít automaticky nakopírované do domovského adresáře nově přidaného uživatele (textový soubor s uvítáním, startovací soubory `.bashrc` a `bash_profile` apod.). Použití skriptu `mass-useradd` je následující:

```
python mass-useradd.py < studenti.txt >> stud_hesla.txt
```

[7] Originální podobu skriptu `mass-useradd.py` lze nalézt na adrese `http://mailman.linuxchix.org/pipermail/techtalk/2004-May/018446.html`.

[8] Protože číslo 101 je již po instalaci rezervováno pro systémovou skupinu `console`

Kromě přidání uživatelů do systému bude výstupem tohoto příkazu také důležitý soubor `stud_hesla.txt` obsahující vygenerovaná hesla pro studenty ve tvaru `jmeno heslo UID`. Ten bude sloužit nejen k distribuci přihlašovacích jmen a hesel studentům před prvním přihlášením, ale také pro usnadnění registrace uživatelů do databáze primárního doménového řadiče NT, o čemž se píše v kapitole 9.

Skript bude nutné spustit i se seznamem učitelů a ostatních uživatelů. V souboru `mass-useradd.py` je však nutné změnit parametr příkazu `useradd` z `-g 102` na odpovídající číslo skupiny právě přidávaných uživatelů a samozřejmě použít příslušný vstupní soubor se seznamem uživatelů.

6.3 Souborové kvóty

Systém kvót umožňuje přidělování kapacity disků jednotlivým adresářům a tím omezuje jejich faktické rozměry. Avšak kvóty nejsou jen jednoduchým omezením diskového prostoru, protože uživatelé jsou omezováni ve svém počínání hned několika způsoby. Proto jsou důležité tyto pojmy:

Uživatelská kvóta – Určuje omezení pro *jednotlivé uživatele*. Právě uživatelskými kvótami se budeme zabývat.

Skupinová kvóta – Určuje omezení pro *skupinu uživatelů*. Obyčejně se nepoužívá a ani jí na serveru používat nebudeme.

Soft limit – Představuje v podstatě „nezávaznou mez“, kterou může uživatel dočasně překročit.

Hard limit – Pevná mez, kterou nelze nikdy překročit. Jinými slovy, uživateli se nikdy nepodaří na disk uložit více, než je uvedeno v tomto parametru.

Inodes, Blocks – Představuje počet souborů, který může uživatel na disk uložit. Tento parametr také nebude na našem serveru využíván, protože omezení ve formě objemu dat je efektivnější a dostačující.

Grace period – Uživatel může dočasně uložit na disk více, než je uvedeno v parametru *soft limit* na dobu zadanou parametrem *grace period*. Po uplynutí této doby se uživateli nepodaří na disk uložit více, i když ještě nepřekročil mez zadanou parametrem *hard limit*.

Aby bylo možné kvóty používat, je třeba mít linuxové jádro přeložené s příslušnou podporou a nainstalovaný software `quota`, který dovoluje nastavovat a ovládat všechny funkce z uživatelského prostoru. Distribuce Slackware obojí obsahuje: použité distribuční jádro má podporu kvót v podobě modulu a balík `quota-3.12-i486-1` se nachází ve skupině `/ap` prvního instalačního CD Slackware Linuxu 10.2.

V případě serveru `orpheus` bude kvótami omezen diskový prostor, kde mají uživatelé své domovské adresáře, tedy oddíl `/home`[9]. Ke správné funkčnosti souborových kvót pro uživatele musí být tento oddíl připojený s volbou `usrquota`, která bude proto uvedena v parametrech připojení oddílu `/home` v souboru `/etc/fstab`. Druhým krokem je vytvoření prázdného konfiguračního souboru `aquota.user` v kořenovém adresáři připojovacího bodu, kam se kvótovaný disk připojuje (tedy v adresáři `/home`). Jeho přístupová práva je nutné nastavit na `600`, aby se souborem nemohli manipulovat obyčejní uživatelé. Pro projevení změn je třeba systém restartovat nebo provést instalaci ručně příkazy:

```
quotacheck -avu
quotaon -avu
```

Program `quotacheck`, který zjistí velikost souborů uživatelů a připraví vše ostatní pro používání kvót, může protestovat, že systém souborů není připojený pouze pro čtení a doporučí ještě parametr `-m`. Tomu lze zabránit přepnutím se do jednouživatelského režimu příkazem `init 1` ještě před spuštěním příkazu `quotacheck`. Použité parametry obou příkazů jsou následující:

`-a` – Zkontroluje, resp. zapne všechny kvóty, jak jsou uvedeny v souboru `/etc/fstab`.

`-v` – Zobrazuje zprávy během práce.

`-u` – Pracuje s uživatelskými kvótami (standardní chování).

Od této chvíle bude startovací skript `/etc/rc.d/rc.M` spouštět potřebné příkazy `quotacheck` a `quotaon` při každém startu systému. Proto je dobré si tento skript prohlédnout a zkontrolovat, zda vyhovují všechny parametry zmíněných příkazů – v našem případě je zde navíc parametr `-g` inicializující skupinové kvóty, které však nevyužíváme.

[9]Uživatelé mají právo zápisu také do oddílů pro dočasné soubory`/tmp` a `/var/tmp`, které by však neměly být pro ně omezené, avšak pravidelně kontrolované a promazávané.

6.3.1 Nastavení kvót pro uživatele

V tuto chvíli kvóty běží, ale zatím ještě nejsou definovány žádné limity. Jak již bylo řečeno v odstavci 6.1, budou využívány modeloví uživatelé (které již máme vytvořené) pro stanovení kvót pro uživatele skutečné. Pro editaci limitů se používá příkaz `edquota` následovaný jménem uživatele, pro kterého bude editace provedena:

```
edquota prototyp_studenti
edquota prototyp_ucitele
edquota prototyp_ostatni
```

Ve spuštěném editoru lze nastavit maximální počet souborů (část `inodes`) nebo jejich celkovou velikost. U obou limitů lze nastavit jak *soft limit*, tak i *hard limit*. V našem případě bude dostačující stanovit následující hodnoty pro část `blocks` – hard:

153600 (= 150 MB) pro `prototyp_studenti`

204800 (= 200 MB) pro `prototyp_ucitele`

51200 (= 50 MB) pro `prototyp_ostatni`

Nyní již zbývá jen aplikovat nově vytvořené limity prototypových uživatelů na všechny uživatele skutečné. Obecný příkaz pro tuto akci je `edquota -p prototyp uzivatele`. Protože musí být kvóty aplikované pro všechny uživatele (kterých je přes 500), bylo by opět velice zdlouhavé je přidávat každému zvlášť. Využijeme tedy tento jednoduchý skript `kvoty.sh`, který aplikuje kvóty pro zadané uživatele v jednom kroku:

```
#!/bin/bash
#
for i in ‘cat /etc/passwd | awk -F":" ’BEGIN { ORS = " " } \
$3 >= 1003 && $3 < 1454 { print $1 }’‘
do
   edquota -p "$1" $i
done
```

Tento konkrétní skript předpokládá, že `UID` prvního skutečného uživatele je 1003 a `UID` posledního se stejnou požadovanou kvótou je 1453. Jako parametr při spuštění je mu

předán název prototypového uživatele, jehož kvóty se mají aplikovat pro uživatele ze zmíněného intervalu čísel `UID`. Například tedy `./kvoty.sh prototyp_studenti`. Tento skript `kvoty.sh` je nutné spustit tolikrát, kolik prototypových uživatelů chceme aplikovat. Vždy se však musí změnit interval čísel `UID` uvnitř skriptu a předat správný parametr při spouštění.

6.3.2 Shrnutí

Každý uživatel může sledovat stav svých kvót pomocí příkazu `quota`, který jej informuje o aktuálních limitech a o jejich naplnění.

Kvóty jsou velmi účinnou zbraní proti neposlušným uživatelům, kteří by chtěli na společné disky nahrát více, než je zdrávo. Zároveň by však správce neměl uživatele omezovat víc, než je nezbytné. Někteří uživatelé a administrátoři nejsou velkými zastánci souborových kvót, protože diskový prostor je dnes levný, a dokonce argumentují, že kvóty nadělají více problémů než jich řeší. Uvědomíme-li si však, že uživateli serveru `orpheus` budou žáci základní školy, kteří navíc mají neomezený přístup k internetu, jsou souborové kvóty opravdu nezbytnou náležitostí.

7 Nastavení sítě

7.1 Sítě TCP/IP

Nedílnou součástí síťového softwaru jsou síťové protokoly. Ty definují komunikační pravidla, jimiž se řídí výměna dat v síti. Pro správnou funkci sítě je tedy nutné, aby všechny síťové stanice používaly stejný protokol. V operačních systémech Linux/Unix, MacOS, Windows a ve většině ostatních se nejčastěji používá síťový protokol TCP/IP (*Transmission Control Protocol / Internet protocol*). Protokol TCP/IP je zároveň přirozeným jazykem internetu. Zařízení ovládající tento protokol si mohou vyměňovat údaje navzdory mnoha rozdílům mezi nimi. Protokol IP je „pracovním“ koněm internetu operující v síťové vrstvě modelu ISO/OSI a jeho úkolem je vysílání datagramů[10] na základě adres v nich obsažených. Je protokolem nespojovým, příjem paketů neověřuje. TCP vytváří spolehlivou službu nad protokolem IP (potvrzuje příjem dat). Udržuje spojení a realizuje konverzaci mezi dvěma programy, přičemž spojení trvá, i když jedna strana „nehovoří“.

[10] Což je v podstatě malý blok dat, který je přenášen z jednoho počítače na jiný prostřednictvím sítě.

UDP je služba soustřeďující se na datagramy. Je podobná jako odeslání dopisu poštou. Neposkytuje obousměrné spojení, nemá žádnou formu prevence ucpání linky a nezaručuje dodání datagramů ve stejném pořadí, v němž byly odeslány. TCP je zdvořilý protokol nutící konkurenční uživatele dělit se o kapacitu linky a obecně se chovat způsobem, který je produktivní pro celou síť. UDP naopak vysílá datagramy tak rychle, jak to jde.

Adresace počítače v sítích TCP/IP je uskutečňována pomocí IP adresy, což je číslo reprezentované čtveřicí třímístných desítkových čísel, přičemž jednotlivé trojice jsou od sebe oddělené tečkou. Každé desítkové číslo může nabývat maximální hodnoty 255. K popisu adresy TCP/IP patří ještě maska, která určuje, kolik bitů z IP adresy patří síti a kolik připadá na hostitele. Máme-li například adresu 147.230.156.97 a síťovou masku 255.255.255.0, tak víme, že prvních 24 bitů tvoří adresu sítě (147.230.156.) a hostitel v této síti má číslo 97. Pokud není síťová maska definována, je adresa rozdělena podle starých tříd adres. Podle Hunta [5, s. 68] tato pravidla říkají následující:

- Jestliže je první byte menší než 128, použij prvních osm bitů pro síť a zbývajících 24 bitů pro hostitele.
- Jestliže je hodnota prvního bytu mezi 128 a 191, použij prvních 16 bitů pro síť a zbývajících 16 bitů pro hostitele.
- Jestliže je hodnota prvního bytu mezi 192 a 223, použij prvních 24 bitů pro síť a zbývajících osm bitů pro hostitele.

7.2 Ethernet

Pro konfiguraci sítě není nic zásadnějšího než rozhraní, které systém používá pro připojení do sítě. Typu hardwaru, který se dnes nejčastěji používá v lokálních sítích, říkáme *Ethernet*. Tento nejrozšířenější standard sítí LAN od poloviny 70. let vyvíjela firma Xerox a dnes existuje více jeho variant. V lokálních sítích se Ethernet prosadil v 80 % všech instalací. Jeho popularita spočívá v jednoduchosti protokolu a tím i snadné implementaci i instalaci. Mezi základní znaky Ethernetu patří kolizní přístupová metoda CSMA/CD, což je přístupová metoda k médiu, při které zařízení chtějící vysílat, čeká na přítomnost mezery v konverzaci (tedy okamžiku, kdy na kabelu neprobíhá komunikace). Je-li mezera detekována, zařízení může vysílat. Jestliže dvě zařízení začnou vysílat ve stejný okamžik, nastává kolize, která je detekována všemi zúčastněnými stanicemi. Stanice pokoušející se

o vysílání, se na jistou náhodně definovanou dobu odmlčí a poté vyšle další pokus o vysílání. Tím lze předejít situaci, kdy by dvě zařízení zároveň vysílala data do sítě, zjistila kolizi, čekala stejnou dobu a znovu začala vysílat, a tak zahltila síť kolizemi.

Jednou z nevýhod ethernetové technologie je omezená délka kabelu, což vylučuje jeho použití pro jiné než lokální sítě. Nicméně pomocí opakovačů, mostů a směrovačů může být pospojováno několik ethernetových segmentů.

V případě Ethernetu lze použít různé topologie a kabely. Díky jeho rozšířenosti je příjemné velké množství aktivních prvků, které jsou na trhu k dispozici. Ethernet existuje ve čtyřech fyzických provedeních:

Tlustý Ethernet – Jeho základem byl tlustý koaxiální kabel, který se lišil šířkou a způsobem, jakým k němu šlo hostitele připojit.

Tenký Ethernet – Tenký koaxiální kabel přímo propojující komunikující zařízení.

Kabeláž optickým vláknem – Používají se jednovidová i mnohavidová vlákna v závislosti na požadované rychlosti a vzdálenosti.

Kabeláž kroucenou dvoulinkou (Twisted pair) – Základem je kroucená dvoulinka, což je odvozenina od telefonního kabelu, a dnes je nejrozšířenějším vodičem v sítích LAN. Síť má hvězdicovou topologii, jejím jádrem je tedy koncentrátor – *hub* nebo *switch*. Hub je pouhý opakovač, který každý přijatý rámec pošle na všechny své porty a neví nic o topologii sítě, zatímco switch (přepínač) si udržuje tabulky s fyzickými adresami a může tak rámce předávat jen na port, na němž je připojen adresát. Provoz switchů je bezpečnější a zajištěna je vyšší propustnost sítě. Rychlost může být 10 Mbit/s, 100 Mbit/s i 1000 Mbit/s, je však podmíněna určitou kvalitou provedení kabeláže

7.3 Aplikace Ethernetu ve školní síti

Pro popisovanou školní síť je důležitý pojem 100BASE-TX. Jedná se druh Ethernetu s přenosovou rychlostí **100** Mbit/s, které se říká Fast Ethernet. Pracuje na kabeláži s nestíněnou kroucenou dvoulinkou (tedy **T**wisted pair) kategorie 5 s využitím dvou párů a maximální délka segmentu může být 100 metrů. Všechny počítače v popisované učebně VT2 jsou propojeny právě kroucenou dvoulinkou a síťovým switchem Eidmax s 24 porty a navzájem se „vidí“.

7.4 Konfigurace ethernetového rozhraní na serveru orpheus

Ethernetové rozhraní Linuxu se skládá z hardwarového zařízení a softwarového ovladače. Server `orpheus` je osazen dvěma ethernetovými kartami Intel s čipem i82559 pro Fast Ethernet. Obecně lze říci, že softwarem ethernetového rozhraní je ovladač jádra, buď zakompilován přímo do něj nebo načten jako zaveditelný modul. Použité distribuční jádro řady 2.4 používá ovladač zmíněných adaptérů Intel právě jako zaveditelný modul. V systému Linux jej lze nalézt v adresáři `/lib/modules/‘uname -r‘/kernel/drivers/net`. Jedná se konkrétně o moduly `e100` a `eepro100`. Linuxový systém detekuje ethernetový hardware a během instalace nainstaluje tyto ovladače. Bývá to automatický proces a proto budeme předpokládat, že žádný zásah administrátora pro instalaci síťových karet není potřeba. Korektní detekci síťových zařízení lze zkontrolovat příkazem `dmesg | grep eth`, který vypíše relevantní záznamy z jádra Linuxu.

Aby byl `orpheus` rozpoznatelný v síti, je potřeba jeho síťovým rozhraním přiřadit konfigurační hodnoty TCP/IP. Toho lze obecně docílit příkazem `ifconfig`, avšak takto definované hodnoty by nepřežily další bootovací proces. Aby bylo síťové rozhraní nakonfigurováno při každém spuštění systému, musí být příkaz `ifconfig` uložen v nějakém spouštěcím souboru. Již při instalaci Slackware Linuxu systém detekuje síťové rozhraní a zeptá se na konfigurační informace týkající se sítě. Ty jsou uloženy na disk a později použity příkazem `ifconfig`. Slackware ukládá tyto informace do souboru `/etc/rc.d/rc.inet1.conf`, který však bude nutné dále upravit, aby `orpheus` komunikoval jak s vnější, tak i vnitřní sítí. Počítač má dvě síťová rozhraní – jedno pro vnější síť, druhé pro vnitřní, se jmény `eth0` a `eth1`. Počítač je přímo připojený k internetu přes rozhraní `eth0` a má v rámci sítě přidělenu veřejnou IP adresu. Pro potřeby této práce bude mít tato adresa hodnotu **172.16.1.31**[11]. Veškeré nezbytné údaje pro připojení serveru `orpheus` k internetu, které jsou přidělené poskytovatelem připojení, uvádí tabulka 5[12]. Kromě vnější sítě je nutné, aby `orpheus` komunikoval také se sítí vnitřní, tedy se všemi žákovskými počítači v učebně VT2. K tomu bude používat své druhé síťové rozhraní, `eth1`, a hlásit se do privátní sítě 192.168.1.0 pomocí adresy 192.168.1.1. Konfigurační soubor `/etc/rc.d/rc.inet1.conf` bude tedy vypadat takto:

```
# Konfiguracni informace pro eth0:
```

[11] Toto je pouze příklad. Ve skutečnosti tato adresa nemůže být použita pro směrování dat v internetu. Je to speciální síťová adresa, která je rezervována pro použití v privátních sítích.

[12] Opět se jedná pouze o příklady. Skutečné hodnoty by musely být odlišné.

IP adresa	172.16.1.31
Síťová maska	255.255.255.0
Brána	172.16.1.250
DNS 1	172.16.2.3
DNS 2	172.16.2.4

Tabulka 5: Síťové informace přidělené poskytovatelem připojení

```
IPADDR[0]="172.16.1.31"
NETMASK[0]="255.255.255.0"
USE_DHCP[0]=""
DHCP_HOSTNAME[0]=""

# Konfiguracni informace pro eth1:
IPADDR[1]="192.168.1.1"
NETMASK[1]="255.255.255.0"
USE_DHCP[1]=""
DHCP_HOSTNAME[1]=""

GATEWAY="172.16.1.250"
```

Aby se projevily změny, je třeba síť restartovat, například příkazy:

```
/etc/rc.d/rc.inet1 stop
/etc/rc.d/rc.inet1 start
```

Skript `/etc/rc.d/rc.inet1` použije k nastavení sítě hodnoty ze svého konfiguračního souboru `rc.inet1.conf`. Správné nastavení lze zkontrolovat příkazem `ifconfig` bez parametrů. Posledními nezpracovanými hodnotami jsou adresy jmenných serverů, které se v Linuxu zapisují do souboru `/etc/resolv.conf`:

```
root@orpheus:~$ cat /etc/resolv.conf
nameserver 172.16.2.3
nameserver 172.16.2.4
```

Síť je v tuto chvíli nastavena a `orpheus` by měl dostat odezvu od jakéhokoliv počítače v internetu pomocí příkazu `ping` *hostitel* (pokud má příjemce povoleny ICMP dotazy).

8 Služba DHCP

Jak uvádějí Droms a Lemon [1, s. 3], protokol pro dynamickou konfiguraci hostitelů (*Dynamic Host Configuration Protocol*) automatizuje proces konfigurace nových i existujících zařízení v sítích TCP/IP. Je to tedy nástroj k poskytování všech možných konfiguračních parametrů svým klientům. Stejně jako jeho starší a jednodušší předchůdce *Bootstrap Protocol* (BootP) pracuje přes UDP porty 67 a 68. Kromě výběru IP adresy na správné podsíti DHCP umožňuje přidělování i dalších informací. Jsou jimi například:

- Adresy směrovačů příslušejících síťovému segmentu, k nimž byl klient připojen
- Adresy doménových serverů
- Masky podsítě a adresy všesměrového vysílání (broadcast)

Jednou z důležitých vlastností služby DHCP je princip *zapůjčování*. Podle Dromse a Lemona [1, s. 9] totiž DHCP server nepřiřadí jednoduše každému klientovi IP adresu na libovolně dlouhou dobu, dokud ji klient potřebuje, nýbrž přiřadí klientovi adresu prostřednictvím tzv. zápůjčky. Když doba zápůjčky uplyne, klient je přinucen přidělenou IP adresu přestat používat. Aby se zabránilo zbytečnému propadání doby zápůjčky, které v zásadě zpomaluje veškeré přístupy klienta k sítí, klient musí svou zápůjčku přidělené IP adresy obnovit ještě předtím, než doba zápůjčky uplyne. Většina klientů DHCP obnovuje zápůjčky mnohokrát po sobě.

Obecně existují dvě zásady přidělování adres. Statické přidělování a dynamické přidělování.

Statické přidělování (neboli *pevné*) pracuje na principu jednoznačné identifikace klientů DHCP. Server DHCP musí obdržet informace, konkrétně identifikující každého klienta a stanoví IP adresu, která mu bude přidělena. Typem identifikátoru, jenž se k tomu dá použít, může být například fyzická (MAC) adresa síťové karty klienta DHCP.

Dynamické přidělování. Server DHCP má k dispozici rozsah IP adres pro každý síťový segment, na kterém má probíhat dynamické přidělování. Když klient DHCP požádá

o přidělení IP adresy, server vyhledá volnou adresu ze správného segmentu a přidělí ji klientovi.

Samozřejmě je možné obě zásady kombinovat a některým klientům v síti přidělovat adresu dynamicky, zatímco u některých (nemobilních) použít statické přidělování.

8.1 Server DHCP ve školní síti

Zřízení serveru DHCP je výhodné i v tak malých sítích jako je ta v učebně VT2. Zkušenosti říkají, že nakonfigurování jednoduchého serveru DHCP pro jedinou podsíť (což je právě náš případ) nezabere více času, než konfigurace síťových parametrů na jednom či dvou klientských počítačích. Na serveru `orpheus` budeme využívat dynamického přidělování IP adres pro jednu podsíť pomocí DHCP serveru od konsorcia ISC v jeho nejnovější verzi 3.0.3, kterou lze stáhnout na adrese `http://www.isc.org/products/DHCP`. Po instalaci by měl být k dispozici prázdný konfigurační soubor `dhcpd.conf` v adresáři `/etc`. Ten bude potřeba upravit, aby přiděloval námi požadované informace, přičemž po všech potřebných úpravách by měl mít následující podobu:

```
# dhcpd.conf
# Configuration file for ISC dhcpd (see 'man dhcpd.conf')
#
authoritative;
#
max-lease-time 604800;
default-lease-time 43200;
option subnet-mask 255.255.255.0;
option domain-name-servers 172.16.2.3, 172.16.2.4;
option routers 192.168.1.1;
option netbios-name-servers 192.168.1.1
#
subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.10 192.168.1.34;
}
# end
```

První nezakomentovaný řádek s hodnotou `authoritative` říká, že DHCP server je autoritativní a jím poskytované informace jsou přesné. Hunt [5, s. 330] říká, že pokud je z nějakého důvodu server konfigurován někým, kdo nemá oprávnění k síťové konfiguraci, může se použít příkaz `not authoritative`, aby se omezila autorita, kterou server uplatňuje na klientech. Avšak server DHCP by měl být vždy autoritativní – a měl by ho mít na starosti pouze síťový administrátor. Což je také případ serveru `orpheus`, takže není důvod direktivu `authoritative` vynechávat. Hodnota `604800` u parametru `max-lease-time` určuje maximální délku zápůjčky IP adresy bez ohledu na délku požadovanou klientem. Je stanovená na 604800 sekund, tedy 1 týden. Hodnota dalšího parametru `default-lease--time` je 12 hodin (43200 sekund). Tím je určena výchozí délka zápůjčky, nepožaduje-li klient nějakou specifickou hodnotu. Počítače v učebně se obvykle zapínají kolem osmé hodiny ráno a díky hodnotě 12 hodin by si měly vystačit se stejnou IP adresou po celou dobu vyučování za jeden den. Další tři hodnoty přiřazují masku podsítě, jmenné servery a síťovou bránu (tou bude opět server `orpheus`, který pro tuto funkci později nastavíme). DHCP bude také díky hodnotě `option netbios-name-servers` posílat klientům se systémem Windows adresu jmenného serveru NetBIOS. Díky službám serveru Samba (viz. kapitola 9) to bude opět počítač `orpheus`. Zbytek souboru tvoří definice podsítě 192.168.1.0 a dává klientům k dispozici 25 adres z rozsahu 192.168.1.10 až 192.168.1.34 (parametr `range`). Vzhledem k tomu, že je v učebně 15 počítačů a občas jsou do sítě připojeny i přenosné počítače vyučujících, je tento rozsah plně dostačující.

Poslední věcí, kterou je nutné udělat, je spuštění samotného DHCP serveru. Ten se spouští příkazem `/sbin/dhcpd` (bez parametrů) a pokud bychom jej chtěli zapínat při každém startu systému (což opravdu chceme), je nutné tento příkaz umístit do souboru `/etc/rc.d/rc.local`. Tabulky výpůjček IP adres lze sledovat v souboru `dhcpd.leases`, který je ve Slackware Linuxu umístěn v adresáři `/var/state/dhcp`.

8.2 Shrnutí

Jak je vidět z výpisu souboru `dhcpd.conf`, je nastavení DHCP serveru pro jednu podsíť velice jednoduchou a přímočarou záležitostí. V případě popisované školní sítě je takto nastavený DHCP server dostačující a skutečně ušetří čas s nastavením síťových informací u všech klientů.

9 Samba

V této části se budu zabývat popisem a nastavením souborového, tiskového a autentizačního serveru Samba. Podle oficiálních stránek tohoto projektu je Samba open source / svobodný software, který poskytuje služby souborového a tiskového serveru SMB/CIFS klientům. Na rozdíl od jiných SMB/CIFS provedení, je možné Sambu používat zcela svobodně a zdarma. Umožňuje spolupráci mezi linuxovými / unixovými servery a klienty s Windows [12].

Samba je tedy softwarový balík unixových nástrojů, který ke komunikaci používá protokol *Server Message Block* (SMB) vyvinutý společnostmi Microsoft a IBM. Operační systémy Microsoft Windows a OS/2 využívají tento protokol k vytváření sítí na bázi klient–server pro sdílení souborů, tiskáren a k vykonávání přidružených operací. Tím, že Samba podporuje protokol SMB, umožňuje unixovým operačním systémům spolupracovat s počítači se systémy Windows na stejné síti, jakoby se jednalo o další klienty s Windows.

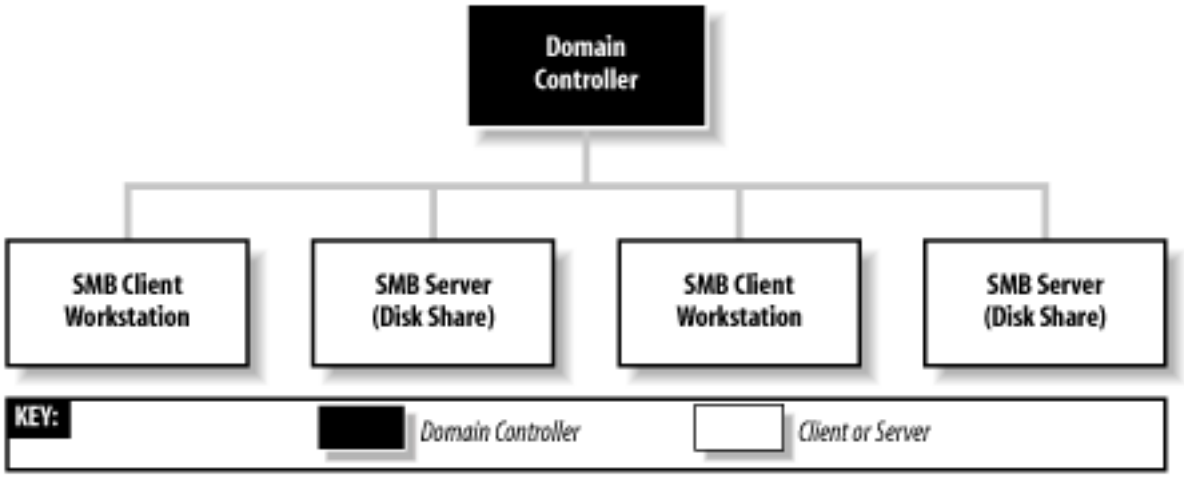
Samba nabízí následující služby:

- Sdílení souborových systémů.
- Sdílení distribuovaných souborových systémů.
- Sdílení tiskáren instalovaných na serveru pro klienty s Windows.
- Podpora klientů při prohlížení síťového okolí.
- Autentizace klientů, kteří se hlásí do domény Windows.
- Podpora převodu jmen WINS (*Windows Internet Name Service*).
- Umožňuje přístup ke sdíleným adresářům a tiskárnám, které v síti nabízejí systémy Windows a další servery Samba.

Server Samba sestává z několika nástrojů a dvou unixových daemonů, které poskytují sdílené prostředky (též nazývaná *sdílení*) všem SMB klientům v síti. Jedná se to tyto daemony:

smbd – Daemon, který se stará o sdílení souborů a tiskáren v síti SMB a zprostředkovává autentizaci klientů.

nmbd – Využívá a poskytuje jmenné služby NetBIOS a WINS, což je implementace jmenného serveru NetBIOS od Microsoftu. Umožňuje také procházení sítí.



Obrázek 3: Schéma jednoduché domény Windows. Zdroj [11].

9.1 Doména Windows NT

Obecně lze říci, že ve školní síti bude Samba využívána pro dvě činnosti: jako tiskový server a primární doménový řadič domény NT. Doména NT (též doména Windows) představuje skupinu počítačů, ve které je server pracující jako *řadič domény*[13] (viz. obr. 3). Microsoft poprvé představil doménu ve svém systému Windows NT 3.51, a proto ji obvykle nazývá „doména NT“ (protože předpokládá, že roli řadiče domény představuje počítač se systémem Windows NT).

Doménový řadič v doméně Windows NT zastává funkci držitele databáze informací o uživatelských účtech a skupinách (stejně jako například služba NIS v sítích UNIX) a vykovává mnohé další služby. Zodpovědnost řadiče domény se vztahuje hlavně k zajištění bezpečnosti, včetně autentizace, což je proces přidělení nebo odepření uživatelského přístupu ke sdíleným prostředkům v doméně. K tomu řadič obvykle používá tzv. *správce zabezpečení účtů (security account manager – SAM)*, který udržuje databázi uživatelů. Řadiči, jenž je v doméně aktivní, se říká *primární řadič domény (primary domain controller – PDC)*. Ten může mít jeden nebo více *záložních řadičů (backup domain controller – BDC)*, které přebírají roli primárního řadiče v době jeho poruchy či nedostupnosti. BDC si pravidelně automaticky synchronizují své databáze SAM s primárním řadičem tak, aby v případě nutnosti mohly vykonávat role řadiče domény, aniž by se tato změna dotkla klientských počítačů. Všechny současné verze Windows mohou přistupovat do domény

[13]Tato vlastnost odlišuje *doménu* od *pracovní skupiny*, ve které se doménový řadič nenachází.

a využívat služeb doménových řadičů. Proč je tedy výhodné hlásit klientské počítače se systémem Windows do domény?

- Centralizovaná správa uživatelských účtů. Jak již bylo řečeno, k přihlášení do domény musí mít uživatel na řadiči vytvořeno své přihlašovací jméno a heslo. Jakmile se počítač takto přihlásí do domény, může jeho uživatel využívat všechny nabízené prostředky platné pro použité uživatelské jméno bez další identifikace.

- Je-li v sítí vytvořena doména, je možné využívat tzv. *cestovní profily* uživatelů. Naprostou samozřejmostí je, že se uživatel může přihlásit z jakéhokoliv počítače v sítí připojeného do domény. S využitím cestovních profilů má navíc k dispozici vždy stejné uživatelské prostředí (které si během práce postupně vytváří), protože jsou jeho data ukládána na server. To zahrnuje vzhled pracovní plochy, obsah nabídky Start a další nastavení, které je uživateli povoleno přizpůsobovat.

- Přihlašovací skripty. Jakmile se klient připojí k doménovému řadiči, je mu poskytnut dávkový soubor MS-DOS (tzv. přihlašovací skript – *logon script*), vytvořený administrátorem a uložený na serveru. Tento soubor obsahuje příkazy, které se spustí při přihlášení do domény a které určují výchozí vlastnosti a podobu prostředí pro přihlášeného uživatele. Výhodou je, že tyto informace mohou být pro každého uživatele v síti jiné.

- Doména Windows nabízí centralizovanou správu sítě. Administrátor může stanovit pravidla ovlivňující pracovní prostředí pro všechny uživatele a přidávat / ubírat práva, jimiž budou uživatelé počítačů v sítí omezováni.

9.2 Active Directory a ti druzí

Se systémem Windows 2000 Microsoft představil databázi síťových objektů s názvem *Active Directory*. Tato implementace je založená na systému adresářových služeb LDAP (*Lightweight Directory Access Protocol*) a využívá jmenných služeb DNS namísto NetBIOS (resp. WINS). Logická struktura Active Directory organizuje všechny prvky databáze podle pravidel, která se snaží kopírovat správní strukturu organizace. Podle Ecksteina [11, s. 34] jsou domény v Active Directory organizovány v hierarchických stromových strukturách, ve kterých každý řadič domény působí na stejné úrovni bez rozlišení funkce primárních a záložních řadičů (jako je to v doménách NT). Podpora Active Directory

v současné verzi Samby (3.0) je značně omezená – Samba může vystupovat pouze v roli klienta.

Sambu lze však nastavit pro využívání samotných adresářových služeb LDAP jako systému pro ukládání informací o uživatelských účtech, resp. samotných uživatelích. O protokolu LDAP pojednává kapitola 12.1.

9.3 Samba jako PDC ve školní síti

Na serveru `orpheus` je klíčové mít instalovánu Sambu ve verzi 2.2 a vyšší. Starší verze totiž nemohou fungovat jako primární doménový řadič pro systémy Windows NT/2000/XP, což je vlastnost, kterou budeme nutně potřebovat, protože na žákovských počítačích bude nainstalován systém Windows XP Professional. Postup popisovaný v mé práci byl vyzkoušen na Sambě ve verzi 3.0.20, což je v době jejího psaní nejnovější stabilní verze.

Klíčem k nastavení serveru Samba je soubor `smb.conf`, který se po jejím nainstalování většinou ukládá do adresáře `/etc/samba`. Příklad souboru `smb.conf` vyhovující nastavení Samby pro popisovanou školní síť tvoří přílohu B této práce.

Konfigurační soubor Samby je rozdělen na globální část (uvozenou klíčovým zápisem `[global]`) a na ostatní části popisující jednotlivá sdílení a další nabízené síťové prostředky. Následující popis vysvětluje jednotlivé konfigurační parametry přiloženého souboru `smb.conf`.

`[global]` začátek globálního nastavení Samby

`workgroup = VT2` pracovní skupina (v tomto případě doména), do které se bude server hlásit.

`netbios name = orpheus` jméno NetBIOS, pod kterým bude Samba vystupovat.

`security = user` určuje systém zabezpečení. Hodnota `user` nastavuje Sambu, aby požadovala uživatelská jména a hesla pro přístup ke sdíleným prostředkům.

`encrypt passwords = yes` tato logická hodnota říká, že pro komunikaci s klientem bude využíván systém šifrovaných hesel. Všechny současné verze Windows jej používají.

`hosts allow = 192.168.1.` určuje, která skupina klientů má právo přistupovat ke sdílenému prostředku. Jestliže je tato hodnota definována v globální sekci, omezení platí

pro všechny sdílení, nehledě na to, zda mají nastavenou jinou hodnotu. Parametru lze udělovat výjimky klíčovým slovem `EXCEPT`.

`log file = /var/log/samba.%m` nastavuje cestu k logovacímu souboru Samby. Ve výsledku bude proměnná `%m` je nahrazena jménem klientského počítače.

`max log size = 50` maximální velikost logovacího souboru v kilobajtech.

`passdb backend = tdbsam smbpasswd` umožňuje vybrat systém, kam se budou ukládat uživatelská jména a hesla Samby. Lze nastavit více možností zadáním parametrů oddělených mezerami. Ty pak budou prohledávány v pořadí, v jakém jsou uvedeny, avšak noví uživatelé jsou přidáváni pouze do prvního z nich. Databáze TDB (volba `tdbsam`) je novější a bezpečnější volbou než pouhé ukládání jmen a šifrovaných hesel do souboru `smbpasswd`.

`socket options = TCP_NODELAY` konfiguruje nastavení socketů, které je specifické pro daný operační systém. Nastavení `TCP_NODELAY` způsobí, že se zasílají pakety libovolné velikosti, aby se snížilo zpoždění.

`interfaces = eth1` nastavuje rozhraní, na kterých bude Samba poslouchat (`eth1` je rozhraní směřující do vnitřní sítě).

`local master = yes` značí, že se server pokusí stát místním hlavním prohlížečem pro danou podsíť. Ještě to však neznamená, že volbu skutečně vyhraje, ale daemon `nmbd` se o to pokusí.

`os level = 65` nastavuje hodnotu příznaku operačního systému, se kterou Samba půjde do volby prohlížeče. Čím vyšší číslo, tím větší je šance na výhru. V popisované školní podsíti není žádný jiný server, který by se o výhru pokoušel a navíc takto vysoké číslo zabezpečí, že Samba volbu vždy vyhraje.

`preferred master = yes` je poslední volbou týkající se voleb místního prohlížeče. S nastavenou logickou hodnotou na `yes` se Sambě zajistí vyšší úroveň než mají ostatní počítače se stejnou hodnotou `os level`. Jestliže chceme systém pro funkci prohlížeče nastavit, měla by tato volba mít hodnotu `yes`.

`domain master = yes` nastavuje Sambu také do role doménového hlavního prohlížeče. Slouží-li Samba jako primární doménový řadič, měla by funkci doménového prohlížeče zastávat také.

`domain logons = yes` nastaví Sambu tak, že bude umožňovat přihlášení do domény jako primární řadič domény NT.

`logon script = logon.bat` specifikuje použití a název přihlašovacího skriptu. Úplná cesta ke skriptu bude definována později.

`logon path = \\%L\profiles\%U` určuje cestu, kam se budou ukládat cestovní profily uživatelů. Skládá se z proměnných a sdílení *profiles*: `%L` (jméno serveru), sdílení *profiles* (bude definováno později) a proměnná `%U` (přihlášený uživatel).

```
add user script = /usr/sbin/useradd %u
```

```
add group script = /usr/sbin/groupadd %g
```

```
delete user script = /usr/sbin/userdel %u
```

```
delete user from group script = /usr/sbin/deluser %u %g
```

`delete group script = /usr/sbin/groupdel %g` těchto 5 operací přidává či ubírá odpovídající unixové uživatele, resp. skupiny. Proměnné `%u` a `%g` jsou nahrazeny jménem uživatele, resp. názvem skupiny.

`add machine script` je skript, který se spustí při pokusu o registraci počítače do domény.

`passwd program = /usr/bin/passwd %u` nastaví cestu k programu, který mění UNIXová hesla uživatelů.

`unix password sync = yes` potvrzuje, že při změně uživatelského hesla v databázi primárního řadiče, bude změněno i unixové heslo daného uživatele.

`time server = yes` zajistí, že se daemon `smbd` představí klientům jako časový server. Samotná synchronizace času mezi serverem a klienty bude nastavena pomocí přihlašovacího skriptu.

`logon drive = H:` specifikuje název jednotky pro mapování domovských adresářů uživatelů v systému Windows.

Tento výpis neobsahuje některé použité volby z globální části přiloženého souboru `smb.conf`. Jedná se o užitečné parametry, ale vždy takové, které nemají zásadní vliv na běh a vlastnosti serveru Samba, a tudíž jejich popis zde není nutný.

Zbývající části konfiguračního souboru již definují jednotlivá sdílení, která jsou vždy uvozená jeho názvem v hranatých závorkách. Většinou se jedná o nastavení vlastností sdílení nějakého obyčejného adresáře, avšak existuje pár sdílení se speciálním významem. První z nich je sdílení s názvem `[homes]`:

```
[homes]
   comment = Domovsky adresar
   browseable = no
   guest ok = no
   writable = yes
```

Pokud je v `smb.conf` definováno sdílení `[homes]`, po přihlášení do domény bude uživateli k dispozici jeho domovský adresář specifikovaný v souboru `/etc/passwd` (v případě popisované sítě je to vždy `/home/uzivatel`). Adresář se připojí jako nová síťová jednotka s názvem odpovídající hodnotě `logon drive` z globální části `smb.conf` a bude omezená souborovými kvótami, které byly nastaveny v kapitole 6.3. Použité nastavení říká, že adresáře nebudou vidět při procházení sítí, bez hesla k nim nepůjde přistupovat a zápis do nich bude možný.

Druhým speciálním sdílením je již zmiňované sdílení `[profiles]`, které slouží jako „úložiště“ cestovních profilů uživatelů:

```
[profiles]
   path = /home/roaming
   browsable = no
   writable = yes
   create mask = 0600
   directory mask = 0700
```

Podle parametru `logon path` z globální sekce souboru `smb.conf` se cesta pro uložení profilu složí z několika proměnných a právě sdílení `[profiles]`. Výsledná cesta bude tedy vždy `/home/roaming/`*uzivatel*. Sdílením nebude možno procházet, ale zapisovat do něj ano. Masky nově vytvořených souborů a adresářů dávají plná práva pro vlastníka, žádná pro všechny ostatní uživatele. Adresář `/home/roaming` je nutné vytvořit ručně a nastavit jeho práva na `777`, aby do něj mohli zapisovat všichni uživatelé. Zde je důležité si uvědomit, že se cestovní profily uživatelů nacházejí v adresáři `/home` a tak se na ně vztahují pro něj nastavené souborové kvóty. Vlastníky souborů z cestovního profilu jsou

samozřejmě ti, komu profil patří, a tak je důležité žáky a učitele upozornit, aby si vždy hlídali zaplnění svých disků.

Posledním speciálním sdílením je adresář, kde je umístěn skript `logon.bat`:

```
[netlogon]
   comment = Network Logon Service
   path = /etc/samba/.winlogon
   guest ok = no
   writable = no
   browsable = no
```

O významu přihlašovacích skriptů se již psalo v kapitole 9.1, a tak jej můžeme rovnou vytvořit. Zde je výhodné si uvědomit, jaké adresáře budeme vlastně sdílet a mapovat na síťové jednotky pomocí přihlašovacího skriptu tak, že budou všem rychle dostupné přes `Tento počítač`. Navrhované řešení by mělo dobře posloužit výuce v učebně:

1. Domovské adresáře uživatelů. Kromě souborů dle vlastního uvážení budou do nich žáci také ukládat odpovědi na různé testy, práce a samostatné projekty. Domovské adresáře se do logon skriptu zadávat nemusejí, jelikož se již mapují díky sdílení `[homes]`.

2. Adresář pro distribuci materiálů pro výuku. Ten by měl obsahovat další adresáře s názvy jednotlivých tříd, které se budou v učebně vyučovat. Do nich budou mít právo zápisu pouze členové skupiny `ucitele`.

3. Sdílení adresáře `/tmp`, které bude přístupné zápisu pro všechny uživatele a bude sloužit k výměně nejrůznějších dat mezi jednotlivými počítači. Připomínám, že pro adresář `/tmp` je vytvořen samostatný diskový oddíl, takže nehrozí nechtěné zaplnění diskového prostoru serveru `orpheus` neposlušnými uživateli.

4. Adresář s instalačními soubory dostupný pouze pro čtení. Ten bude sloužit ke sdílení instalačních souborů a ovladačů, které bude nutné nainstalovat na všech stanicích. Jejich sdílení pro všechny počítače v učebně tento proces velmi urychlí.

Podle tohoto postupu je nyní nutné jednotlivá sdílení vytvořit, což znamená upravit soubor `smb.conf`:

```
[vyuka]
   comment = Soubory pro vyuku
   path = /home/share/vyuka
   public = no
   printable = no
   write list = @ucitele
   force group = ucitele

[verejne]
   comment = Verejne sdileni
   path = /tmp
   read only = no
   public = no

[install]
   comment = Instalacni soubory
   path = /home/share/install
   read only = yes
   public = no
```

Pro sdílení `vyuka` je třeba vytvořit adresář `/home/share/vyuka`, změnit skupinu na `ucitele` a přidat jí práva zápisu (`chmod g+w /home/share/vyuka`). Parametry v nastavení sdílení `[vyuka]` způsobí, že adresář bude přístupný pro zápis skupině `ucitele` a budou jí také patřit všechny nově vytvořené soubory. Sdílení s názvem `[verejne]` je přístupné všem i pro zápis, zatímco sdílení `[install]` je dostupné pouze pro čtení.

Takto vytvořená sdílení budou dostupná při prohlížení sdílených prostředků na serveru `orpheus`. Námi nadefinovaný logon script však způsobí, že se všechna tato sdílení budou připojovat jako nové síťové jednotky po přihlášení uživatele do domény. Logon skript je v podstatě dávkový soubor MS-DOS s příponou `.bat` (nebo `.cmd`), takže jej vytvoříme. Do prázdného souboru `logon.bat` v adresáři `/etc/samba/.winlogon/` je nutné napsat tyto řádky:

```
net time \\orpheus /set /yes
net use I: \\orpheus\install
```

```
net use T: \\orpheus\verejne
net use V: \\orpheus\vyuka
```

Řádky začínající klíčovými slovy `net use` způsobí připojení jednotlivých sdílení a nastavují písmeno jednotky. První řádek v souboru synchronizuje čas na stanicích s časem na serveru[14]. Protože se bude dávkový soubor `logon.bat` spouštět na systémech Windows, je nutné aby by napsán ve formátu MS-DOS, tedy s odpovídajícími konci řádků. Toho lze docílit několika způsoby. Jedním z nich je například editace souboru v editoru `vim` a jeho přeformátování příkazem `:se ff=dos`. Další možností je použití utility `unix2dos`: `unix2dos logon.unix > logon.bat`.

9.4 Registrace uživatelů do Samby

Samba používá vlastní databázi k uchování kombinací uživatelské jméno / heslo (nevyužívá tedy databázi Linuxu, soubor `/etc/passwd`). Do databáze má samozřejmě právo zápisu pouze doménový administrátor, kterým je v Sambě uživatel `root`. Je však důležité vědět, že se jedná o jiného správce `root` než je ten linuxový. K přidání uživatelů do databáze Samby slouží příkaz `smbpasswd` s parametrem `-a`:

```
smbpasswd -a root
```

Hodnota parametru `passdb backend` v souboru `smb.conf` říká, že se k uchování uživatelů bude primárně užívat databáze TDB (trivial database, hodnota `tdbsam`). Ta je novější a bezpečnější náhradou výchozí databáze `smbpasswd`. Databáze TDB implicitně ukládá své soubory do adresáře `/etc/samba/private`, přičemž ke skutečným uživatelům se váže soubor `private.tdb`. S přidáním prvního uživatele (tedy doménového administrátora `root`) se soubor `private.tdb` vytvoří a uživatel bude do něj zaznamenán.

Nyní je čas přidat všechny zbývající uživatele do databáze TDB (jak již bylo řečeno, je jich zhruba 500)[15]. Jejich postupná registrace příkazem `smbpasswd -a` by opět byla časově náročná, a tak lze využít skript `addsmb.sh`, který tvoří přílohu C. K jeho správné funkčnosti je nutné předat mu jako parametr při spouštění soubor se jmény a hesly uživatelů, který byl dříve vytvořen jako výstup skriptu `mass-useradd.py`. Připomínám, že

[14] Časová synchronizace je důležitá pro správnou funkčnost cestovních profilů.

[15] Je důležité vědět, že se musí jednat o uživatele, kteří mají na serveru skutečně zřízený účet s platným UID.

ten má na každém řádku informace o jednom uživateli ve tvaru `login␣heslo␣UID`. Skript prochází tento soubor po řádcích a postupně spouští příkaz `smbpasswd` (v neinteraktivním režimu) pro každého uživatele a nastaví mu tak stejné uživatelské jméno a heslo jako má daný uživatel v databázi Linuxu. Pro kontrolu je možné skriptu `addsmb.sh` předat jako druhý parametr název souboru, kam má zaznamenat svůj vlastní výstup ve tvaru `samba_login␣samba_heslo`.

Kromě samotných uživatelů je nutné do Samby zaregistrovat i všechny počítače, které budou vstupovat do vytvořené domény. K tomuto účelu slouží příkaz `add machine script` z přiloženého souboru `smb.conf`. Pokaždé, když nějaká stanice požádá o registraci do domény, Samba zkontroluje, zda k tomu má oprávnění (podle restrikcí v `smb.conf`) a využije příkaz z parametru `add machine script` k registraci stanice do své databáze. Konkrétně se jedná o soubor `secrets.tdb` ve stejném adresáři, kde je uložen `private.tdb`. Tento proces je zcela automatický, jen je důležité vědět, že počítače do domény může přidávat pouze doménový administrátor.

9.5 Samba jako tiskový server

Kromě autentizace uživatelů přihlašujících se do nově vytvořené domény Windows a sdílení adresářů, se bude Samba využívat také jako tiskový server. V učebně VT2 je k dispozici USB tiskárna HP LJ 2200, která bude po správném nastavení přístupná všem 15 žákovským počítačům, zobrazí se v seznamu sdílených prostředků serveru `orpheus` a bude možné ji k cílovému počítači jednoduše připojit.

Před samotným nastavením Samby ke sdílení tiskárny pro klienty SMB je samozřejmě nutné, aby tiskárna na serveru bezchybně fungovala. V Linuxu existuje několik odlišných tiskových systémů, z nichž jeden je nutné využít pro nainstalování a spravování tiskárny. V současnosti je asi nejoblíbenějším tiskovým systémem CUPS *(Common Unix Printing System)*, jehož administrace probíhá velice jednoduše pomocí webového prohlížeče. Běží-li tiskový daemon `cupsd`, stačí do prohlížeče napsat adresu `http://127.0.0.1:631/` (localhost, port 631) a v přehledném prostředí vykonávat všechny činnosti spojené s administrací tiskáren a tiskových úloh. Ať už se použije CUPS či starší linuxový tiskový systém LPD (resp. LPRng), pro Sambu je důležité, aby výsledkem byl platný konfigurační soubor tiskáren `/etc/printcap`.

Jak uvádí Eckstein [2, s. 194], zaslání tiskové úlohy na server Samba je složeno ze čtyř kroků:

1. Otevření a autentizace připojení k tiskovému sdílení
2. Přenos souboru po síti
3. Odpojení
4. Tisk a odstranění kopie ze souboru

Když Samba přijme tiskovou úlohu, data, která se mají tisknout, jsou dočasně uložena na disk do adresáře specifikovaném volbou `path` v příslušném tiskovém sdílení. Samba poté provede unixový příkaz pro zaslání tištěných dat na tiskárnu. Tisková úloha je vytištěna a připsána autentizovanému uživateli. Pokud je tak tiskové sdílení nastaveno, může tímto uživatelem být i `guest`.

Pro sdílení tiskárny přes Sambu existují dvě možnosti. Za prvé můžeme v globální části souboru `smb.conf` definovat sekci `[printers]`. Pokud tak učiníme, Samba automaticky načte obsah souboru, který definuje nastavení tiskáren (obvykle to bývá `/etc/printcap`, pokud nedefinujeme jiný parametrem `printcap name`), a pro každou tiskárnu v něm nalezenou vytvoří samostatné tiskové sdílení. Druhá možnost spočívá v explicitním vytvoření samostatného sdílení pro každou tiskárnu v systému, přičemž pomocí parametru `printable = yes` Samba pozná, že se jedná o sdílení tiskové. Využít tuto druhou možnost se vyplatí, chceme-li například povolit tisk pouze několika konkrétním uživatelům. Jejich jména by se v tomto případě uvedla k parametru `valid users` u příslušného tiskového sdílení. Nicméně ve školní síti bych preferoval spíše první možnost, protože ulehčuje správu tiskáren při jejich případné výměně či přidání druhé tiskárny. Bude-li tedy někdy v budoucnu k serveru připojena druhá tiskárna (a tedy i upraven konfigurační soubor `/etc/printcap`), Samba pro ní automaticky vytvoří tiskové sdílení, a tak bude možné tiskárnu ke stanicím rychle připojit bez dodatečné konfigurace souboru `smb.conf`.

Kromě samotného připojení tiskárny na klientském počítači, je jediným nutným krokem k jejímu nasdílení úprava souboru `smb.conf`:

```
[global]
load printers = yes
printing = cups
.
.
[printers]
```

```
comment = Tiskarny na serveru
path = /var/spool/samba
guest ok = no
writable = no
printable = yes
```

Hodnotou `load printers = yes` docílíme toho, že všechny tiskárny zapsané v konfiguračním souboru `/etc/printcap` budou přístupné při prohlížení sítě. Jako tiskový systém byl vybrán CUPS. V sekci `[printers]` definujeme cestu k dočasnému ukládání tiskových úloh, tisk neautorizovaným uživatelům není dovolen a jakákoliv modifikace sdílení je zakázána. Příkaz `printable = yes` zde musí být explicitně uveden.

Poznámky k tisku

1. Aby bylo vůbec možné na tiskárně tisknout, je důležité si uvědomit, že UNIX považuje všechny tiskárny za PostScriptové, a tudíž je potřeba mít v systému nainstalovánu takovou tiskárnu, jejíž procesor umí PostScript použít.[16] Pokud to tak není (což je právě případ tiskárny HP LJ 2200), je možné tuto činnost „simulovat“ programem Ghostscript, což je interpretr jazyka PostScript. Ghostscript bývá součástí grafického prostředí Linuxu (které však na serveru `orpheus` není), avšak lze jej nainstalovat i samostatně.

2. Pokud je k nainstalování a spravování tiskárny použit systém CUPS, je nutné provést dvě změny v jeho konfiguračních souborech, aby bylo možné tisknout ze systému Windows (který používá tzv. *RAW tisk*). Je tedy nutné povolit (odkomentovat) řádek `application/octet-stream` v souboru `/etc/cups/mime.convs` a ten samý v souboru `/etc/cups/mime.types`.

3. Nastavení parametru `printing` v globální části souboru `smb.conf` například na hodnotu `BSD` (tedy jiný systém tisku) umožňuje definovat příkaz, který má Samba vykonat pro zaslání tiskového souboru na tiskárnu (parametrem `print command`). Tak si lze například vést jednoduchou statistiku vytištěných dokumentů:
   `print command = echo "Tisk %f na %p" >> /tmp/printlog; /usr/bin/lpr -P%p -r %s`
   Tento příkaz zaznamená informace o tištěné úloze do souboru `/tmp/printlog` a

[16] PostScript je v podstatě jazyk popisující stránku, vyvinutý společností Adobe.

provede její tisk příkazem `lpr`. Proměnné `%p` a `%f` budou nahrazeny jménem tiskárny, resp. jménem souboru, který se má tisknout (proměnná `%s` bude navíc obsahovat úplnou cestu k tomuto souboru na serveru Samba). Při nastavení systému tisku na CUPS se parametr `print command` ignoruje.

Takto nasdílenou tiskárnu je nyní možné jednoduše připojit na všechny klienty v síti, kteří komunikují přes protokol SMB. Pomocí utility `smbclient` lze tiskárnu využívat i na počítačích se systémem Linux. Konfigurace klientů, včetně připojení systému Windows do domény NT je popsána v kapitole 15.1.

10 Spolupráce s Linuxem

V předcházející kapitole jsem popisoval, jak žákovským počítačům, resp. jejich uživatelům zpřístupnit síťové prostředky nabízené serverem `orpheus`. Byla však řeč pouze o klientech, na kterých je nainstalován systém Windows. V kapitole 3.1 je uvedeno, že na všech žákovských počítačích bude možno zavést jakýkoliv ze dvou systémů – Windows nebo Linux. Cílem této kapitoly tedy bude nastavit `orpheus` tak, aby stanice mohly jeho služby využívat také v době, kdy na nich bude spuštěn systém Linux. Nejprve se budu snažit uživatelům zpřístupnit jejich domovské adresáře a poté nastavit centralizovanou správu jmen a hesel pro systém Linux, přičemž pro tuto funkci popíšu 3 způsoby, kterými lze dosáhnout téměř stejného výsledku.

10.1 Síťový souborový systém

Síťový souborový systém (NFS) umožňuje sdílet souborové systémy mezi počítači. Uživatelé pak přistupují k souborům umístěným na vzdálených systémech, jako by se jednalo o soubory lokální. Jak říká Hunt [5, s. 361], NFS je systém typu klient/server. Klienti používají vzdálené adresáře tak, jako by byly součástí místního systému souborů; server umožní jejich použití. Zpřístupnění vzdáleného adresáře lokálním systémem souborů se nazývá *připojení* (mount) a zpřístupnění adresáře ostatním se říká *export*.

Systém NFS běží nad protokolem RPC (vzdálené volání procedur) firmy Sun, který definuje systémově nezávislý způsob, jakým mezi sebou mohou komunikovat počítače v síti. Jako přenosový protokol pro NFS může být použit jak UDP tak i TCP, nicméně podpora TCP je stále považována za experimentální. Standardní chování NFS spočívá v tom, že

daemoni, kteří mají na starosti NFS požadavky, nemají přiřazena stálá čísla portů UDP, ale ty jsou jim dynamicky přidělována programem RPC portmapper. Pro úspěšný export adresářů musí tedy tento program běžet a vzhledem k tomu, že on sám používá standardní číslo portu (111), můžou ho vzdálené počítače kontaktovat. Portmapper pak přiřadí čísla portů dalším daemonům NFS, z nichž klíčové jsou `rpc.mountd` a `rpc.nfsd` zpracovávající požadavky klientů a poskytující skutečné souborové služby.

Samotná definice adresářů, které se mají exportovat (tedy nabízet klientským počítačům) se provádí v souboru `/etc/exports`. Abychom dosáhli podobného výsledku, jakým bylo mapování domovských adresářů v systému Windows pomocí Samby, bude nutné exportovat celý adresář `/home` ze serveru `orpheus` a připojovat jej ke všem klientským počítačům, také k přípojnému bodu `/home`. Žáci tak budou moci přistupovat ke svým souborům z obou operačních systémů. Soubor `exports` má tento obsah:

```
/home 192.168.1.0/255.255.255.0(rw,root_squash,sync)
```

Je vidět, že adresář `/home` je exportován klientům ze sítě 192.168.1.0/255.255.255.0. Volba `rw` zajistí, že na disk půjde zapisovat, avšak pouze do podadresářů, jejichž přístupová práva to umožňují. Na žákovských počítačích budou totiž mít všichni uživatelé stejné UID a GID jako na serveru `orpheus` a budou mít tedy právo manipulovat pouze se svým vlastním domovským adresářem. Parametr `root_squash` zajistí, že se požadavky (na připojeném adresáři) obsahující UID a GID uživatele `root` na jakékoliv stanici budou mapovat na UID a GID uživatele `nobody`. Poslední parametr `sync` zajistí synchronní I/O přístup k souborům.

Zbývá pouze službu NFS spustit. Startovací skript Slackware Linuxu `rc.nfsd` zajistí start portmapperu i všech obslužných daemonů NFS:

```
/etc/rc.d/rc.nfsd start
```

Správné nastavení exportovaných adresářů lze překontrolovat příkazem `exportfs`. Připojování domovských adresářů do systémů Windows i Linux s sebou nese velké výhody při výuce. Žáci tak mohou pohodlně pracovat se stejnými soubory v obou systémech a poznat tak rozdílné přístupy a výsledky při stejné práci, avšak s odlišnými nástroji (v systému Linux a Windows). Přístup k vlastním souborům je tedy vždy opravdu snadný a neodpoutává žákovu pozornost od vlastního řešení zadané úlohy.

10.2 Autentizace

Zatímco autentizace uživatelů v systému Windows pomocí Samby byla v podstatě jediná možnost, pro ověřování totožnosti v Linuxu máme na výběr více alternativ.

10.2.1 NIS: Síťová informační služba

Služba NIS *(Network Information Service)* vydaná firmou Sun v osmdesátých letech minulého století, byla v té době špičková administrátorská databáze. Je to systém založený na bázi klient / server umožňující zpřístupnění dat ze serveru na klientské počítače. Postupně se ale vyprofiloval do systému služeb, který umožňuje sdílení datových souborů souvisejících s uživateli, přihlašováním, hesly a podobně. V terminologii NISu však není jednotkou sdílení soubor, nýbrž *záznam*, jenž je zpravidla tvořen jedním řádkem konfiguračního souboru. Ten zůstává zachován na svém místě v původním formátu a server tak pomocí záznamů poskytuje jeho obsah ostatním počítačům, které s ním tvoří tzv. *doménu NIS*. Jak uvádí Nemeth, Snyder a Hein [7, s. 463], s každou položkou může být spojen pouze jeden klíč, takže systémový soubor může být přeložen do několika map NIS. Například soubor `/etc/passwd` se překládá do dvou různých map, nazvaných `passwd.byname` a `passwd.byuid` (pro vyhledávání položek podle jména a podle hodnoty UID). První způsob, jak ověřit totožnost přihlašovaných uživatelů v učebně VT2, bude tedy použít službu NIS.

Na serveru NIS (tedy serveru `orpheus`) sídlí datové soubory NIS v adresáři `/var/yp` a každá mapa NIS je uložena v hashovaném formátu v podadresáři nazvaném podle vytvořené domény. Její název se zapíše do souboru `/etc/defaultdomain` a aplikuje příkazem

```
nisdomainname ‘cat /etc/defaultdomain‘
```

V souboru `/var/yp/Makefile` lze zkontrolovat seznam námi požadovaných položek, které zpřístupníme klientům (za hodnotou `all:`) a v souboru `/var/yp/securenets` určit, jakým sítím bude umožněno číst mapy na serveru (zapisuje se ve tvaru s maskou na prvním místě: 255.255.255.0 192.168.1.0, tedy síť v učebně VT2). Samotné sestavení databáze a inicializace serveru se provádí příkazem

```
/usr/lib/yp/ypinit -m
```

Ve Slackware Linuxu již existuje předpřipravený startovací skript služby NIS `rc.yp` v adresáři `/etc/rc.d/`, ve kterém stačí pouze odkomentovat správné řádky a přidat mu práva spouštění, aby se NIS vždy aktivoval při startu systému. Pro promítnutí změn do všech map NIS při každé změně lokálních souborů (`/etc/passwd, /etc/group`, apod.) je nutná aktualizace databáze NIS, která se provádí příkazem `make` v adresáři `/var/yp`. Těmito kroky se vytvoří funkční server NIS, avšak k úplné realizaci této služby je třeba také správně nastavit klientské počítače. To je popsáno v kapitole 15.2.

Je vidět, že zřídit službu NIS pro autentizaci uživatelů je velice snadné, což je její výhoda a jeden z důvodů nasazení ve školní síti. Po korektním vytvoření domény NIS a nastavení přidružených prvků, zbývá správci serveru pouze aktualizovat databázi NIS pokaždé, když je provedena nějaká změna v relevantních konfiguračních souborech. I tuto akci však lze zautomatizovat uvedením příkazu `make` do vlastních skriptů pro přidávání a odebírání uživatelů (viz. kapitola 11). Nevýhodou služby NIS je její malá bezpečnost. Podle Nemetha, Snydera a Heina [7, s. 467] je její vysílací režim zvlášť špatný, protože kterýkoliv počítač na síti může tvrdit, že obsahuje určitou doménu a vnutit klientům NIS falešné údaje. Pro snížení rizika lze však počítačům explicitně vyjmenovat povolené servery NIS a navíc „ten pravý“ přiřazovat dynamicky pomocí serveru DHCP (položkami `option nis-domain` a `option nis-servers` v souboru `dhcpd.conf`).

10.2.2 Využití programů `rsync` a `ssh`

Druhou možností, jak klientským počítačům v učebně poskytnout informace o uživatelích na serveru, je využití kombinace programů `rsync` a `ssh` k synchronizaci uživatelských účtů. To má jednu velkou výhodu: data putují po sítí v šifrované podobě. Idea je opět taková, že všechny změny týkající se uživatelských účtů jsou prováděny pouze na serveru `orpheus` a pro distribuci relevantních souborů je použit program `rsync` spolupracující s interpretem příkazů `ssh` pro bezpečnou komunikaci.

Open source verze programu `ssh` (openssh) by měla být nainstalována na obou stranách a v tomto bodě je důležité, aby její serverová část `sshd` byla spuštěná na stanicích[17]. Ověření totožnosti uživatele přistupujícího na vzdálený počítač přes `ssh` obvykle probíhá zadáním uživatelského jména a hesla. Synchronizace souborů by však měla probíhat bez interakce se správcem, a tak bude nutné k autentizaci použít systém kombinace veřejného a privátního klíče pro uživatele `root`. Příkazem

[17] Podrobnější informace o `ssh` lze nalézt v kapitole 14.1.

```
ssh-keygen -t rsa -b 2048 -C "muj klic" -N ""
```

vygeneruje správce své klíče. Parametrem `-N ""` se nastaví kontrolní fráze na nulovou hodnotu, aby bylo skutečně možné přistupovat ke stanicím bez jakékoliv interakce s uživatelem. Dvojice klíčů se uloží do adresáře `~/.ssh`, z nichž veřejná část (`id_rsa.pub`) musí být nakopírována na stanice a uložena v souboru `~/.ssh/authorized_keys2`. Nesmí se zapomenout také na správné nastavení přístupových práv známým příkazem `chmod` (`chmod 600 .ssh/authorized_keys2; chmod 700 .ssh/`). Od této chvíle může uživatel `root` přistupovat na klientské stanice bez zadání hesla a celá komunikace bude šifrovaná. Obecnou synchronizaci souboru pomocí programu `rsync` lze provést tímto příkazem:

```
rsync -p -e ssh /etc/passwd 192.168.1.25:/etc/passwd
```

Soubor `/etc/passwd` na stanici 192.168.1.25 se synchronizuje se souborem `/etc/passwd` ze serveru bez žádosti o heslo uživatele `root` (parametr `-p` zajistí zachování přístupových práv). Skript `syncpwd.sh` (který tvoří přílohu D) tuto činnost automatizuje tak, že se v zadaném síťovém rozsahu postupně dotazuje klientů, zda běží (příkazem `ping`) a pokud od nich dostane odpověď, provede příkaz `rsync` se zadanými soubory. Spuštění programu `syncpwd.sh` lze opět uvést ve vlastních skriptech pro přidávání a odebírání uživatelů (viz. kapitola 11) a tím jej aktivovat při jakékoliv změně konfiguračních souborů jako jsou `/etc/passwd` a jiné. Nevýhoda spočívá v tom, že pokud budou některé počítače vypnuté (nebo na nich bude spuštěn systém Windows), dojde k synchronizaci pouze s těmi zbývajícími. Řešení lze nalézt ve spuštění programu `rsync` pomocí plánovače úloh `cron`.

10.2.3 Shrnutí

Ověřování uživatelů jak službou NIS tak i za pomocí programu `rsync` a `ssh` má své výhody i nevýhody. Druhá možnost je sice bezpečnější, protože komunikace probíhá přes kryptovaný kanál, ale zase neumožňuje uživatelům na klientských počítačích měnit svá hesla. Každá taková změna by byla totiž ztracena již při další synchronizaci souborů ze serveru `orpheus`[18]. Z hlediska výuky je pravděpodobně lepší použít autentizaci pomocí služby NIS, protože nijak neomezuje žáky ani učitele při probírání učiva o příkazu `passwd`.

[18] V tomto případě je určitě vhodné odstranit spouštěcí příznak z programu `passwd` na stanicích (`chmod 644 /usr/bin/passwd`).

11 Přidávání nových uživatelů

V předcházejících kapitolách jsem popisoval, jak do systému hromadně přidat několik desítek uživatelů najednou, což bylo nutné pro počáteční instalaci serveru. Měl by však existovat také způsob, jak do fungujícího systému přidávat nové uživatele jednotlivě. V případě serveru `orpheus` tato akce vyžaduje několik kroků:

1. Získání přihlašovacího a celého jména uživatele, volba hesla a skupiny, do které patří.
2. Zaznamenání uživatele do databáze Linuxu (včetně vytvoření domovského adresáře).
3. Zaznamenání uživatele do databáze Samby.
4. Nastavení souborových kvót (podle skupiny, do které má uživatel patřit).
5. Aktualizace databáze NIS nebo aktivace programu `rsync` pro synchronizaci souborů na stanicích.

Samozřejmě je možné postupně spouštět jednotlivé příkazy, které tyto kroky uskuteční, avšak mnohem pohodlnější je seskupit je do jednoho vlastního programu, který po získání potřebných informací provede všechny nezbytné kroky sám. Příklad takového programu (`newuser.sh`) tvoří přílohu číslo E. Pátý krok je v programu `newuser.sh` realizován aktualizací databáze NIS, je-li však pro autentizaci uživatelů vybrán druhý způsob (viz. kapitola 10.2.2), je nutné program `newuser.sh` v tomto kroku přizpůsobit. Výstup skriptu může mít například následující podobu:

```
root@orpheus:~# ./newuser.sh

*** Pridani noveho uzivatele ***
* Zadej prihlasovaci jmeno: kamil
Toto jmeno jiz existuje! Vyberte jine.
* Zadej prihlasovaci jmeno: michal
* Zadej cele jmeno uzivatele: Michal Novak
* Zadej heslo:
* Zadej heslo znovu:
```

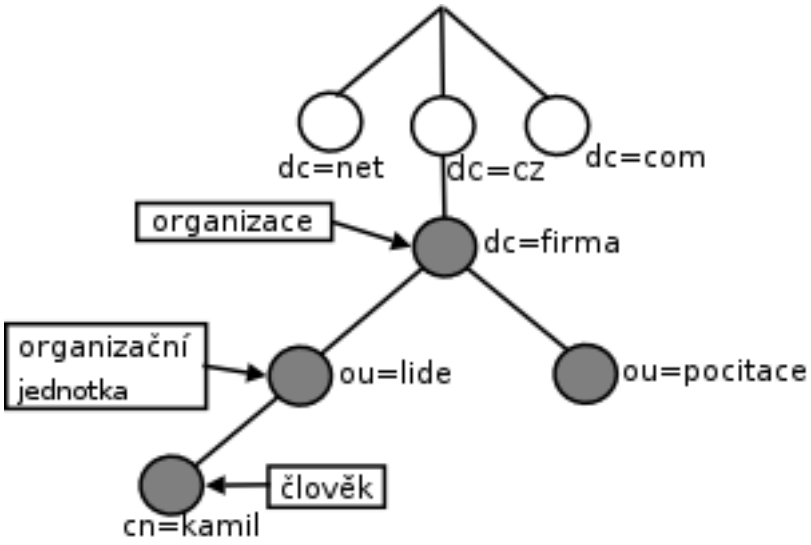
```
Hesla nejsou stejna!
* Zadej heslo:
* Zadej heslo znovu:
* Zadej skupinu: pracovnici
Tato skupina neexistuje, zadejte jinou!
* Zadej skupinu: ucitele

+++ Zadane prihlasovaci jmeno: michal
+++ Zadane cele jmeno: Michal Novak
+++ Zadana skupina: ucitele

Zpracovavam uzivatele...

** Uzivatel michal (uid 1011) pridan do unixove databaze **
** Domovsky adresar /home/michal byl nastaven **
** Uzivatel michal pridan do databeze Samby **
** Diskova kvota pro uzivatele michal byla nastavena **
** Databaze NIS byla aktualizovana **
```

Tento, ani žádný podobný program však není nutné používat, pokud bude k autentizaci uživatelů použita adresářová slušba LDAP, popisovaná v následující kapitole.

12 Adresářové služby

Adresářovou službou se rozumí specializovaná aplikace pro ukládání dat, jejich organizaci a přístup k nim. Všechna data jsou uložena ve formě záznamů obsahující atributy a jejich hodnoty. Každý záznam má unikátní jméno a také každý atribut je v rámci konkrétního záznamu jednoznačně pojmenovaný. Logicky jsou takto položky v adresářových službách rozmístěny v hierarchické struktuře, v tzv. adresářovém stromu. V souvislosti s tím, se setkáváme se pojmem DIT (Directory Information Tree), který představuje konkrétní návrh struktury adresářového stromu, tj. jeho větvení, členění položek a nesených informací do hierarchicky uspořádaných skupin. Příklad jednoduchého adresářového stromu je zobrazen na obrázku 4.

Je důležité si uvědomit, že struktura stromu se přímo projeví ve jménech jednotlivých



Obrázek 4: Jednoduchý adresářový strom

položek. Jméno každé položky je složeno z části popisující, do které větve adresářového stromu položka patří. DN (Distinguished Name) je rozlišovací jméno jednoznačně identifikující položku v globálním jmenném prostoru adresářového stromu. DN se skládá z RDN (Relative Distinguished Name), tzv. relativního rozlišovacího jména, specifikujícího položku v rámci jedné větve stromu. Z tohoto hlediska lze říci, že takováto hierarchická architektura a způsob pojmenování jednotlivých položek je analogická k souborovému systému a jeho adresářové struktuře.

Každý atribut má své jméno a každé položce je přiřazen typ (objectclass), určující, které atributy se mohou v položce vyskytovat. Na obrázku 4 je specifikována podvětev, která je umístěna v oblasti pojmenované dle organizace, v našem případě je to `dc=firma,dc=cz`. V této větvi jsou umístěny všechny objekty (položky) adresářového stromu na uvedeném obrázku. Znamená to, že všechna jména budou mít výše uvedenou specifikaci jako příponu (suffix). Pojmenování, neboli rozlišovací jméno (DN) položky, uživatele *kamil* na obrázku je tedy `cn=kamil,ou=lide,dc=firma,dc=cz`. (cn v řeči adresářových služeb znamená commonName a ou je organizationalUnit).

12.1 LDAP

LDAP *(Lightweight Directory Access Protocol)* je síťový protokol pro přístup k adresářovým službám, jejich ukládání a organizaci, pracující nad internetovým transportním protokolem TCP/IP. Původně byl navržen jako brána, umožňující TCP/IP klientům přistupovat a komunikovat s adresářovými servery X.500. Postupně však došlo k přepracování koncepce LDAP (osamostatnění) a dnes se pod tímto pojmem rozumí nejen komunikační protokol pro přístup k datům, ale i adresářový server samotný.

12.2 LDAP ve školní síti

Hlavní výhodou nasazení LDAP ve školní síti je centralizovaná databáze uživatelských jmen a hesel, ke které mohou přistupovat oba operační systémy na žákovských počítačích. Odpadá tedy práce s udržováním dvou oddělených systémů pro ukládání dat a jejich vzájemná synchronizace. Operační systém Linux má přístup k informacím uložených v databázi LDAP přímo přes balík nástrojů `ldap-client` a po správném nastavení serveru Samba budou tyto informace k dispozici i systému Windows.

Protokol LDAP implementovala firma Netscape a další subjekty na univerzitě ve státě Michigan. V současnosti je nejlepším zdrojem informací skupina OpenLDAP, která převzala a vylepšila původní zdrojový kód. Program OpenLDAP se dodává s mnohými distribucemi nebo lze stáhnout na adrese `http://www.openldap.org`. Instalovat se bude opět na server `orpheus`. Během instalačního procesu se systém dotáže na několik zásadních informací, včetně administrátorského hesla či hodnoty „základního názvu pro LDAP“ – přípony *suffix*[19]. Srdcem konfigurace serveru LDAP je soubor `/etc/ldap/slapd.conf`. První změna (nutná pro správnou funkčnost autentizace pomocí Samby) je umístění schématu Samby na server `orpheus` a přidání jeho záznamu za ostatní schémata pomocí direktivy `include`. Schémata obsahují specifikaci tříd objektů a atributy typů, které chceme aby náš server LDAP podporoval. Soubor `samba.schema` lze nalézt ve zdrojovém kódu příslušné verze Samby.

```
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema
```

[19] Ta bude mít pro potřeby této práce hodnotu `dc=kolej,dc=vslib,dc=cz`

```
include         /etc/ldap/schema/samba.schema
```

Po několika implicitních zápisech v souboru `slapd.conf` (které se nebudou měnit) je nutné specifikovat náš suffix a rozlišovací jméno (DN) správce:

```
suffix   "dc=kolej,dc=vslib,dc=cz"
rootdn   "cn=admin,dc=kolej,dc=vslib,dc=cz"
```

Další nutností je zabezpečit změnu hesel. Následující záznam říká, že atribut heslo uživatele – `userPassword` půjde změnit pouze jeho vlastníkem, pokud byl úspěšně autorizován. Ostatní, kromě správce jej nemohou ani vidět:

```
access to attrs=userPassword
        by dn="cn=admin,dc=kolej,dc=vslib,dc=cz" write
        by anonymous auth
        by self write
        by * none
```

Dalším bezpečnostním záznamem (který je umístěn až za všemi předešlými, protože se při jejich vyhodnocování postupuje shora) je ten, který povoluje správci zápis do všech atributů, ostatním pouze čtení:

```
access to *
        by dn="cn=admin,dc=kolej,dc=vslib,dc=cz" write
        by * read
```

Dalším krokem je úprava konfiguračního souboru Samby `/etc/samba/smb.conf`. Ta nyní musí využívat LDAP jako systém pro ukládání uživatelských jmen a hesel.

```
passdb backend = ldapsam:ldap://127.0.0.1
ldap suffix = dc=kolej,dc=vslib,dc=cz
ldap admin dn = "cn=admin,dc=kolej,dc=vslib,dc=cz"
ldap delete dn = No
ldap machine suffix = ou=Machines
ldap group suffix = ou=Groups
ldap user suffix = ou=People
ldap passwd sync = Yes
```

Hodnoty všech těchto direktiv jsou zřejmé. Direktiva `ldap delete dn = No` říká, že při požadavku na odstranění se smaže pouze atribut patřící Sambě, nikoliv celý záznam.

12.3 Konfigurace klienta LDAP

Pokud se pro autentizaci uživatelů v síti využívá LDAP, není v podstatě nutné je přidávat do databáze `/etc/passwd` tak, jak bylo dříve popsáno. Při konfiguraci klientských počítačů stačí každý z nich uvědomit, že má pro autentizaci využívat službu LDAP na vzdáleném počítači (viz kapitola 15.2). Pokud bychom však chtěli uživatelům umožnit například vzdálený přístup na server `orpheus`, či z nějakého jiného důvodu povolit uživatelům přístup na něj (například v budoucnu přístup FTP), bude nutné `ldap-client` nastavit i na `orpheus` samotný.

Pokud příslušný klient LDAP využívá pro autentizaci uživatelů systém zásuvných modulů do jádra PAM (Pluggable Authentication Modules), bude nutné kromě balíků `ldap-client`, `libnss-ldap` nainstalovat také `libpam-ldap`, tedy podporu LDAP pro moduly PAM. Balík `libnss-ldap` dokáže povolit podporu služby LDAP při nastavování priorizace zdrojů administrátorských informací (tedy i autentizačních) v souboru `/etc/nsswitch.conf`. Prvním krokem ale bude editace souboru `/etc/ldap.conf` a přidání záznamů o samotném LDAP:

```
HOST    127.0.0.1
BASE    dc=kolej,dc=vslib,dc=cz
```

Při instalaci balíku `libnss-ldap` se nás instalátor zeptá na potřebné informace, které je však později možné doplnit do souboru `/etc/libnss-ldap.conf`:

```
host 192.168.1.1
base dc=kolej,dc=vslib,dc=cz
ldap_version 3
```

Potom už je možné přidat záznam `ldap` do souboru `/etc/nsswitch.conf` tak, aby byla definována správná vyhledávací cesta pro relevantní administrátorské informace:

```
passwd: files ldap
group: files ldap
shadow: files ldap
```

Kromě *souborů* (tedy `/etc/passwd` apod.) se bude při autentizaci klientů využívat i informací v LDAP. Poslední editace připadá v úvahu, pokud systém využívá zmíněné moduly PAM. Klíčový zápis `auth sufficient pam_ldap.so` je nutné přidat do souborů

definující příslušné moduly PAM. Jedná se o soubory `/etc/pam.d/common-account`, `/etc/pam.d/common-auth` a `/etc/pam.d/common-password`.

Poslední věcí je vytvoření administrátorského hesla pro server Samba, což se v případě využívání LDAP vytvoří příkazem:

```
smbpasswd -w nove_heslo
```

To je uloženo do souboru `secrets.tdb` a je nutné jej aktualizovat pokaždé, když se změní hodnota `ldap admin dn` v souboru `smb.conf`. Zbývá pouze spustit deamona `slapd` a restartovat Sambu.

12.4 Vkládání a změna údajů LDAP

Vkládání údajů do databáze LDAP probíhá přes tzv. soubory LDIF. Formát takového souboru je však poněkud složitý. Část LDIF souboru definujícího uživatele `tomas` může vypadat například takto:

```
dn: cn=tomas,ou=People,dc=kolej,dc=vslib,dc=cz
uid: tomas
cn: tomas
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
userPassword:: e2NyeXB0fXg=
loginShell: /bin/bash
uidNumber: 206
gidNumber: 10000
homeDirectory: /home/tomas
```

Je zřejmé, že vytváření podobných souborů pro všechny uživatele a jejich následná registrace příkazem `ldapadd` by byla velice dlouhá. Existují však i jiné cesty, jak záznamy o uživatelích registrovat. Firma Padl Software poskytuje bezplatně sadu skriptů v jazyce Perl, které provádějí migraci obyčejných souborů nebo map NIS na LDAP. Lze je získat na adrese `http://www.padl.com/OSS/MigrationTools.html`.

Zajímavým projektem je také LDAP Account Manager. Ten je psán v jazyce PHP a umožňuje velice jednoduše spravovat uživatelské účty, skupiny, počítače či domény NT přes přehledné webové rozhraní.

12.5 Závěr

LDAP je bezpochyby výborná služba, která ve školní síti ušetří udržování dvou databází jmen a hesel a jejich synchronizaci. Po konfiguraci serveru LDAP je možné spravovat veškeré záznamy pomocí webového rozhraní, což velice ulehčuje administrátorské úkony. Udržování záznamů o uživatelích v databázi LDAP je navíc „investice do budoucnosti“, protože se takto uložené záznamy dají využít pro indentifikaci a zobrazování uživatelských informací ve spoustě služeb. Například přístup k webovému obsahu tak může být omezen pouze pro uživatele s platným účtem v LDAP, díky modulu `mod_ldap` http serveru Apache.

13 Proxy server `squid`

Jedním z dalších požadavků školy je zajistit, aby žáci nemohli přes internetový prohlížeč přistupovat na webové stránky s nevhodným obsahem. Jedna z možností jak toto zajistit, je použít k filtrování internetového obsahu tzv. *proxy server*, což je program, který funguje coby prostředník mezi dvěma sítěmi. Jeho úkolem je naslouchat na „kraji“ vnitřní sítě, přijímat a odpovídat na požadavky na komunikaci se sítí vnější a zároveň tyto požadavky zpracovávat podle stanovených pravidel. Jinými slovy, místo toho aby se klientský počítač spojil přímo s cílovým zdrojem, je nucen kontaktovat proxy server, který požadovanou informaci získá, zpracuje a pošle zpět klientovi. Kromě toho je schopen již získanou odpověď uložit (do obsahové cache paměti) a předat dalšímu klientovi, který se v zadaném čase pokusí získat stejnou informaci. Tím se předávání zrychlí a zároveň se i značně ušetří kapacita linky.

Nejznámějším (a patrně také nejlepším) proxy serverem na platformě Linux je `squid`. Jedná se o velice propracovaný nástroj, u něhož vyjmenované vlastnosti proxy serverů tvoří pouze zlomek jeho skutečných schopností. Ve školní síti se samozřejmě všechny jeho možnosti nevyužijí, ale zároveň se s ním dosáhne přesných požadavků na počítačovou učebnu ve škole. Dost často zastává funkci proxy serveru a směrovače tentýž fyzický počítač, a tak bude `squid` nainstalován opět na server `orpheus`.

`Squid` je jedním z nástrojů, které se nenacházejí v základní instalaci Slackware Linuxu, a tak bude třeba jej stáhnout v binární podobě, popřípadě zkompilovat ze zdrojových kódů. Ty lze nalézt na oficiální stránce `http://www.squid-cache.org/`, případný binární balíček pak na adrese `http://www.linuxpackages.net`.

13.1 Ukládání webového obsahu

První věcí, kterou budeme na nově nainstalovaném proxy serveru nastavovat, je ukládání webového obsahu do cache paměti. Konfigurační soubor `squidu` – `squid.conf` se většinou nachází v adresáři `/etc/squid`. Spousta výchozích a vyhovujících hodnot je v něm již zaznamenaná, avšak zakomentovaná, a tak bude stačit je aktivovat, případně také změnit.

Minimální formát konfiguračního souboru pro `squid` s aktivním ukládáním do paměti cache musí obsahovat tyto položky:

http_port 3128 určuje port, na kterém bude `squid` běžet. Port 3128 je standardní číslo pro proxy servery.

cache_mem 8 MB Velikost fyzické paměti, kterou `squid` použije navíc jako cache v paměti.

maximum_object_size 4096 KB je maximální velikost jednoho objektu uloženého v cache paměti.

cache_dir ufs /var/lib/squid/cache 1400 16 256 Nastavuje paměť cache. Použitý typ `ufs` je nativní formát pro `squid`, následuje cesta k adresáři pro ukládání souborů, jeho maximální velikost v MB a počet podadresářů ve dvou úrovních.

cache_access_log /var/lib/squid/logs/access.log je cesta logovacího souboru pro informace o jednotlivých požadavcích od klientů.

cache_log /var/lib/squid/logs/cache.log Tento log obsahuje informace o běhu programu.

cache_store_log /var/lib/squid/logs/store.log je logovací soubor pro informace o vykonávané práci s cache.

Takto nastavený `squid` je připravený pro ukládání webového obsahu. Zmínil jsem však, že uložené informace v paměti cache se klientům posílají pouze po určitou dobu a pak jsou obnoveny, aby se nepřistupovalo k neaktuálnímu obsahu serverů. Toto rozhodování `squid` provádí tzv. *refresh algoritmem*, který se řídí informacemi z HTTP hlaviček přijatých objektů. Ty mimo jiné obsahují čas načtení z HTTP serveru, čas poslední změny a čas vypršení. Aktuálnost informací je obnovovacím algoritmem vyhodnocována podle předpokladu, že se objekty mění s určitou pravidelností. To znamená, že pokud již byl určitý

objekt stažen velmi starý (podle času poslední změny), je pouze malá pravděpodobnost, že se v nejbližší době změní, a tak zůstává v cache uložen déle. Naopak, pokud uživatel stáhl objekt chvíli poté, co byl naposledy změněn, je pravděpodobné, že se bude měnit často. Chování algoritmu lze ovlivnit parametrem `refresh_pattern` v konfiguračním souboru `squid.conf`, ale za normálních okolností vždy postačí výchozí hodnoty.

Před samotným spuštěním proxy serveru je třeba vytvořit pracovní adresáře pro cache a také všechny ostatní potřebné soubory (například logovací). To za nás provede příkaz `squid -z`. Pak již stačí `squid` spustit příkazem `squid -D` a přidat jej do inicializačních skriptů, aby se spouštěl při každém startu serveru `orpheus`.

13.2 Filtrování internetového obsahu

K filtrování internetového obsahu, tedy zamezení přístupu na webové stránky s nevhodným obsahem, se využije systém ACL *(Access Control List)* proxy serveru `squid`. ACL je systém, který umožňuje kontrolovat a zpracovávat požadavky klientů na základě nejrůznějších vlastností, které mají odeslané žádosti. Pomocí vlastností, jako jsou IP adresa odesílatele, cílová adresa, čas, kdy je požadavek vznesen, port apod., lze definovat třídy požadavků a ty různě zpracovávat.

K filtrování obsahu internetových stránek, na které se bude v učebně přistupovat, lze použít systém klíčových slov, které má obsahovat URL cílové stránky. Pokud bude obsahovat slovo (nebo pouze část z něj), které je uvedeno v předem definovaném seznamu zakázaných (čili nevhodných) slov, bude přístup na stránku odepřen a místo ní, zobrazena vlastní varovná zpráva. Pod předem vytvořené a doporučené (recommended) třídy v souboru `squid.conf` je třeba přidat vlastní třídy požadavků, které mohou mít například tento tvar:

```
acl nevhodne url_regex -i "/etc/squid/nevhodne.txt"
acl vhodne url_regex -i "/etc/squid/vhodne.txt"
```

Klíčovým slovem `acl` začíná definice tříd `nevhodne`, resp. `vhodne`. Vlastnost `url_regex -i` porovnává regulární výraz v celé adrese URL se vzory uvedenými v textových souborech `nevhodne.txt` a `vhodne.txt` (parametr `-i` značí, že nezáleží na velikosti písmen). Poté lze již definovat parametr *http_access*, který rozhodne, co se stane s požadavkem odpovídající dané třídě.

```
http_access deny nevhodne !vhodne
```

Slovem `deny` zakážeme „průchod“ třídou `nevhodne` a povolíme třídu `vhodne` (vykřičník totiž značí negaci). Tento příklad předpokládá, že v souboru `nevhodne.txt` máme uveden seznam slov, které by se neměly vyskytnout v URL cílové internetové stránky. Ten můžeme buď postupně vytvářet sami, nebo stáhnout nějaký předpřipravený z internetu. Dostatečně dlouhý a spolehlivý seznam klíčových slov dělá tento systém skutečně efektivním. Je však třeba myslet také na slova o stejném základu. Máme-li například v souboru `nevhodne.txt` uvedeno klíčové slovo *sex*, byl by uživatelům odepřen přístup také na stránky, jejichž URL má v názvu slova *msexcel* nebo *essex*. Tomu lze zabránit vytvořením zmíněné třídy `vhodne` a přípustná slova vyjmenovat v souboru `vhodne.txt`. Tato práce opět nemusí být starostí systémového správce, protože seznamy „vhodných“ slov lze na internetu také nalézt. Komplexní seznamy nabízí například „černá listina“ výrazů, kterou lze stáhnout na adrese `http://www.squidguard.org/blacklist/`.

Vytvoří-li se několik tříd zpracovávající různá nevhodná témata (drogy, pornografie, warez, násilí apod.), stává se systém ACL skutečně sofistikovaným pomocníkem pro filtrování internetového obsahu v počítačové učebně ve škole.

Poslední věcí, kterou je nutné v souboru `squid.conf` upravit, je povolení přístupu k internetovému obsahu z naší sítě (klíčové slovo `src`). Upozorňuji, že *http_access* vztahující se k této třídě, musí být v seznamu ACL uveden až na posledním místě, tedy po vyhodnocení všech předchozích pravidel:

```
acl vt2 src 192.168.1.0/24
http_access allow vt2
```

13.3 Shrnutí

`Squid` je velmi šikovný program, jehož využití ve školní sítí je bezesporu přínosné. Přináší nejen zrychlení práce na lince, ale také zabezpečení přístupu pouze na stránky s vhodným obsahem pro žáky základní školy. Třídou `acl msie browser MSIE` lze také úplně zakázat používání Microsoft Internet Exploreru, který žáci často používají, ale přes který do systému Windows proudí spousty škodlivého softwaru.

14 Bezpečnost

Zabezpečení počítačové sítě je výsledkem bezpečností politiky, která se skládá z několika obranných linií, jež obklopují celý systém a chrání jej před nejrůznějšími útoky. Bezpečnost je tedy základem provozování spolehlivé počítačové sítě. V této kapitole se budu zabývat pouze síťovou bezpečností, ale je nutné vědět, že celková odolnost systému závisí také na dalších faktorech:

- Fyzické zabezpečení počítačů, zejména pak síťového serveru, který musí být chráněn proti neoprávněnému přístupu k systémové konzoli. V neposlední řadě je také nutné myslet na běžné fyzikální vlivy, které by mohly naše počítače (resp. v nich uložená data) nenávratně ohrozit.
- Pravidelné aktualizace programů. Ty umožňují opravovat bezpečnostní chyby za chodu systému, a tak jej udržovat stále v aktuálním stavu.
- Školení uživatelů o problematice bezpečných hesel, sociálního inženýrství a celkové bezpečnostní politice sítě. Důležité je také zmínit případné sankce, kterým podlehnou uživatelé porušující bezpečnostní řád.
- Vedení záznamů o důležitých činnostech serveru (logování) včetně pravidelné analýzy logů. Pravidelné zálohování důležitých dat v systému.

14.1 Síťové zabezpečení serveru `orpheus`

Server `orpheus` je přes svoji veřejnou IP adresu přímo připojen k vnější síti, což pro něj (potažmo i pro ostatní počítače v učebně) představuje jisté bezpečnostní riziko. Zároveň ale neposkytuje žádné služby (nebo téměř žádné, viz. dále) počítačům v jiné síti, než je ta v učebně VT2, takže jeho případné napadení se dá pomocí několika kroků minimalizovat tak, že všechny jeho otevřené porty budou přístupné pouze z vnitřní sítě.

Všeobecně se má za to, že prvním krokem ke zvýšení bezpečnosti serveru je vypnutí známých nebezpečných služeb, pokud není z nějakého důvodu nezbytné, aby byly spuštěny. Jedná se o služby *finger* (získávání informací o uživatelích na lokálních i vzdálených systémech), *telnet* a *rlogin* (nešifrované zpřístupnění vzdáleného terminálu), *ftp* (protokol pro přenos souborů – opět nešifrovaně), *DNS* (překlad doménových jmen na IP adresy) a další. S klidem lze konstatovat, že žádná z těchto služeb na serveru `orpheus` neběží a ani není nutné ji spouštět.

Je však pravděpodobné, že školní správce serveru jej bude chtít spravovat vzdáleně, buď z nějakého jiného počítače ve škole, nebo ze svého domácího PC připojeného k internetu. K tomu je nepochybně výhodné použít službu SSH (Secure Shell), bezpečnou alternativu programů pro poskytnutí vzdáleného terminálu. Ta totiž disponuje několika bezpečnostními prvky, které dělají ze vzdálené správy serveru bezpečnou činnost. Umožňuje ověřování identity hostitele, šifrování přenosu pomocí asymetrických šifer (včetně fáze zadávání hesla), a tak zabraňuje odposlechu komunikace a v neposlední řadě je možné vytvořit šifrovaný tunel pro zapouzdření jiného protokolu. Tvorba klíčů na klientské straně již byla popsána v kapitole 10.2.2 a tak stačí říci, že klíče pro serverovou část se generují automaticky při prvním spuštění serveru `sshd`. Veřejná i tajná část klíče je uložena v adresáři `/etc/ssh`, kde se nacházejí i konfigurační soubory jak pro klientskou tak i pro serverovou část `ssh`. V nich se nachází několik voleb, pomocí nichž lze `ssh` nastavit podle svých potřeb. Pro zvýšení bezpečnosti lze například zakázat přímé přihlášení uživatele `root`. K získání jeho práv je potom nutné znát alespoň 2 hesla – jedno pro běžného uživatele a po změně identity příkazem `su` také vlastní heslo správce. Samotné přihlášení na vzdálený systém lze pak uskutečnit příkazem `ssh hostitel`.

14.2 Netfilter

V počítačových sítích počítače komunikují odesíláním a přijímáním datových paketů. Často je nutné tyto pakety přesměrovávat, měnit nebo úplně zastavit, v závislosti na zdroji, který je odeslal, cíli, ke kterému směřují, či na základě dalších informací obsažených v jejich hlavičkách. To umí služba zvaná *Netfilter*. V Linuxu je tento filtr zabudován přímo v jádře a jmenuje se `iptables`. Kromě zmíněných vlastností lze pomocí `iptables` vytvářet firewally k ochraně počítačů včetně možnosti práce s překladem adres *(Network Address Translation, NAT)*. Práce s NAT umožňuje zejména realizaci „maškarády“ (masquearade, překlad adresy na adresu přiřazenou NATu), předávání portů (port forwarding) a přesměrovávání (redirecting).

Ve školní síti budeme `iptables` využívat k těmto činnostem:

1. Překlad privátních adres pomocí NAT pro zajištění konektivity všech žákovských počítačů v učebně k internetu prostřednictvím veřejné IP adresy serveru `orpheus`.

2. Zajištění bezpečnosti všech počítačů v učebně pomocí filtrování příchozích paketů (firewall).

3. Přesměrování všech HTTP požadavků (tedy na port 80) žákovských počítačů na port proxy serveru `squid` (port 3128).

Cílem této kapitoly není podrobně popsat principy práce s filtrem `iptables`, ale s jeho pomocí vytvořit fungující pravidla k zajištění vyjmenovaných činností. Příloha číslo F této práce obsahuje celý skript `firewall.sh` spouštěný při startu serveru. Uvedená pravidla v této kapitole tvoří pouze vybranou část z celého skriptu. Skript `firewall.sh` jsem vytvořil za pomocí vzoru `mpfw` od Miroslava Petříčka, jehož originální podobu lze nalézt na adrese `http://www.petricek.cz/mpfw/mpfw`.

Jádro Linuxu dělí provoz paketů do tří skupin a na každou z nich lze pomocí `iptables` aplikovat různá filtrovací pravidla[20]:

INPUT Příchozí pakety pro daný počítač (většinou pro určitou službu).

OUTPUT Odchozí datový tok, generovaný přímo daným počítačem.

FORWARD Data, která daným počítačem pouze procházejí, nejsou určena pro něj ani v něm nevznikla.

Vzhledem k tomu, že `orpheus` funguje také jako směrovač, budou v pravidlech použity všechny řetězce. Po zavedení potřebných modulů do jádra Linuxu a povolení předávání paketů, je ve skriptu `firewall.sh` obsaženo pravidlo pro ochranu sítě proti podvržení falešných IP adres pomocí jaderné služby `rp_filter`. Pokud je aktivní (obsahuje číslo 1), tak automaticky odmítá všechny příchozí pakety z adres, jejichž záznam ve směrovací tabulce neodpovídá síťovému rozhraní, přes které přicházejí. Výchozí politika všech řetězců je nastavená tak, aby se veškeré pakety, které výslovně nepovolíme, zahazovaly. Pomocí pravidla

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

přesměrováváme všechny pakety směřující přes vnitřní rozhraní (eth1) z portu 80 na port proxy serveru `squid` (3128). Tím bude zajištěna jeho funkčnost a využitelnost (jedná se o tzv. transparentní proxy) a nebude nutné nic měnit a nastavovat na klientské straně. Pomocí posledních šesti pravidel v řetězci PREROUTING zahazujeme pakety přicházející z nedovolených IP adres. Pomocí dalšího pravidla:

[20] Ve skutečnosti existují ještě dva další řetězce (**PREROUTING, POSTROUTING**), přes které pakety proudí ještě před (resp. po) domluvou s místní směrovací tabulkou.

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 172.16.1.31
```

umožňujeme všem dotazům vystupujícím přes vnější rozhraní (eth0) maskování za veřejnou adresu serveru. Tím se zajistí konektivita celé vnitřní sítě k internetu. Pomocí prvních dvou pravidel v řetězci FORWARD zahazujeme procházející pakety, které se snaží navázat nové spojení, ale nemají příslušný příznak (SYN). Poté povolíme směrování ven ze sítě. Další pravidlo

```
iptables -A FORWARD -i eth0 -o eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

udává tzv. stavový firewall, což znamená, že průchod přes vnější rozhraní dovnitř sítě bude povolen pouze již navázaným spojením. Poté začíná řetězec INPUT, v jehož prvním pravidle opět zahazujeme pakety, které se snaží navázat nové spojení, ale nemají příznak SYN. Po zabezpečení proti nechtěnému skenování portů, povolujeme některé ICMP dotazy (Echo Reply, Destination Unreachable, Echo Request, Time Exceeded). Dále je nutné povolit loopback, pakety z vnitřní sítě a broadcasty na lokálním rozhraní. Následujícím pravidlem propustíme pakety od navázaných spojení:

```
iptables -A INPUT -d 172.16.1.31 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

V posledním řetězci OUTPUT je nutné povolit odchozí pakety, které mají naše vlastní IP adresy (veřejnou, všechny lokální a loopback). Poslední pravidlo umožňuje DHCP broadcasty na vnitřním rozhraní, aby bylo možné přiřazovat síťové informace klientům pomocí DHCP serveru.

Firewall s těmito pravidly pro `iptables` nebude do vnitřní sítě z internetu propouštět žádné pakety, kromě zmíněných kontrolních paketů ICMP. Aby však bylo možné využít dálkovou správu serveru pomocí SSH i z jiného počítače, než jsou ty v učebně, je třeba přidat ještě toto pravidlo do řetězce INPUT:

```
iptables -A INPUT -i eth0 -s 1.2.3.4 -p tcp --dport 22 -j ACCEPT
```

Přístup na port 22 (`sshd`) pak bude možný také z IP adresy 1.2.3.4. Ve školním prostředí se také někdy hodí žákům blokovat určité služby, využívající známé porty. Pokud by je žáci pouštěli během výuky, je možné je zakázat na úrovni `iptables`:

```
iptables -A FORWARD -p tcp --dport 5190 -j DROP
```

Toto pravidlo nepovoluje veškerý průchod ze sítě směřující na port 5190, což je port, na kterém běží komunikační protokol ICQ.

Poslední věcí, kterou je nutné v tomto okamžiku udělat, je doladit proxy server `squid`, aby bylo skutečně možné využívat jeho služeb. Tím, že jsme veškerou komunikaci směřující na port 80 přesměrovali, jsme přišli o cílovou IP adresu a proxy tím pádem neví, kam posílat dotazy. Podle Krčmáře [6, s. 165] posílají prohlížeče společně s požadavkem o zaslání obsahu také direktivu Host informující web server o tom, ke kterému virtuálnímu serveru má zájem prohlížeč přistupovat. Tuto direktivu využijeme a proxy pomocí ní určí cílový server.

```
httpd_accel_host virtual
httpd_accel_uses_host_header on
httpd_accel_port 80
```

Přidáním (odkomentováním) těchto tří řádků do konfiguračního souboru `squid.conf` sdělujeme proxy, že má využívat informace nalezené v direktivě Host, pracovat zároveň jako akcelerátor pro web server a určíme číslo portu cíle. Zároveň jsme však v komentáři informováni, že povolením parametru `httpd_accel_host` přijdeme o ukládání webového obsahu do paměti cache. Tuto funkci ale můžeme znovu povolit posledním novým řádkem do `squid.conf`:

```
httpd_accel_with_proxy on
```

14.3 Shrnutí

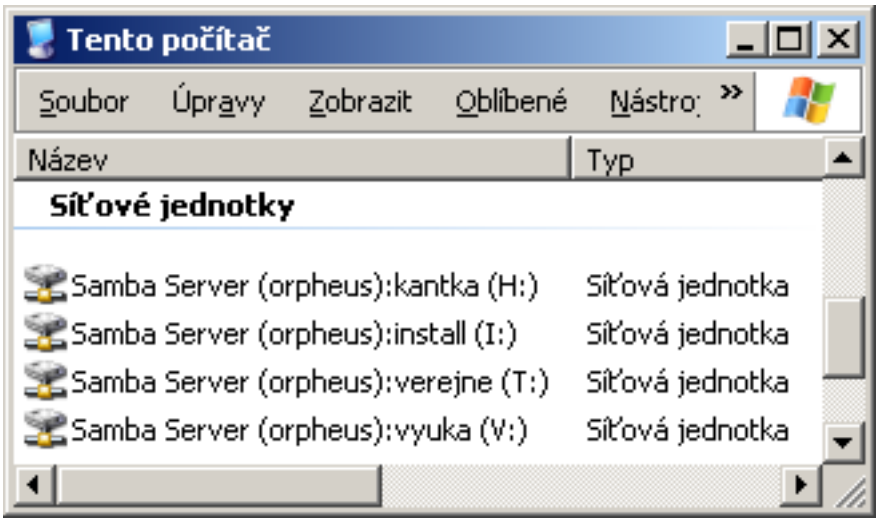
Bezpečnost počítačové sítě je téma, které se nevyplatí podceňovat ani v tak malé síti jako je ta v učebně VT2. Přestože přítomné počítače neobsahují potenciálně atraktivní data, vždy se musí počítat s rizikem, že špatně zabezpečený systém může útočníkům sloužit pouze jako „přestupní stanice“ pro napadení jiných sítí. Vinu za způsobenou škodu na vzdálené straně by pak mohla nést škola.

15 Nastavení klientských počítačů

Po úspěšném nainstalování a konfiguraci všech služeb na serveru `orpheus` bude nutné nastavit žákovské počítače. V této fázi předpokládám, že oba operační systémy jsou na nich korektně nainstalované a propojené v síti tak, jak bylo znázorněno na obrázku 2 v kapitole 3. Zbývá tedy jen klienty nastavit aby využívaly služeb serveru.

15.1 Windows XP Professional

Již při instalaci Windows se systém pokusí získat síťové informace od DHCP serveru. V této fázi lze také počítač přidat do domény NT zadáním jejího jména a identifikačních informací doménového administrátora (připomínám, že se jedná o uživatele `root`). Pokud by byl již systém nainstalován, je nutné změnit jeho identifikaci v síti a přidat jej do domény NT dodatečně (Přes nabídku *Jméno počítače* ve vlastnostech oblasti *Tento počítač*). Předtím je však potřeba v síťovém nastavení povolit získávání síťových identifikačních informací serverem DHCP. Upozorňuji, že tyto kroky může vykonávat pouze uživatel patřící do skupiny Administrators. Při změně identifikace počítačů je výhodné jejich názvy nějak logicky uspořádat (například VT2–1 až VT2–15). Po přihlášení uživatele do domény se připojí všechny síťové disky tak, jak bylo nastaveno na serveru Samba a definováno přihlašovacím skriptem `logon.bat` (viz. obrázek 5).



Obrázek 5: Síťové disky v systému Windows XP

Poslední věcí, kterou je nutné v systému Windows XP udělat, je přidání síťové tiskárny. K tomu je opět nutné přihlásit se k systému jako uživatel s administrátorskými právy a najít tiskárnu ve sdílených prostředcích serveru `orpheus`. V kontextové nabídce tiskárny potom stačí kliknout na příkaz `Připojit...`, vybrat typ tiskárny a umístění ovladačů. Ty je možné mít uložené například na síťovém disku `install`, aby byly rychle přístupné pro všechny počítače v učebně.

15.2 Linux

První otázkou, kterou se musíme zabývat ohledně žákovských počítačů s Linuxem, je jakou distribuci na ně vybrat. V současné době je na stanicích nainstalována distribuce SuSE 9.3, a tak se dá předpokládat, že na ní žáci budou již dostatečně zvyklí. Z hlediska výuky se tato distribuce jeví jako vhodná varianta, protože ve své instalaci obsahuje všechny potřebné programy pro práci s textem i grafikou a zároveň nepostrádá klasické unixové nástroje pro práci v příkazové řádce. Velká vazba na grafické prostředí a množství grafických ovládacích prvků v tomto případě nejsou na škodu a z počátku výuky operačního systému Linux je jistě vhodné je žákům představit.

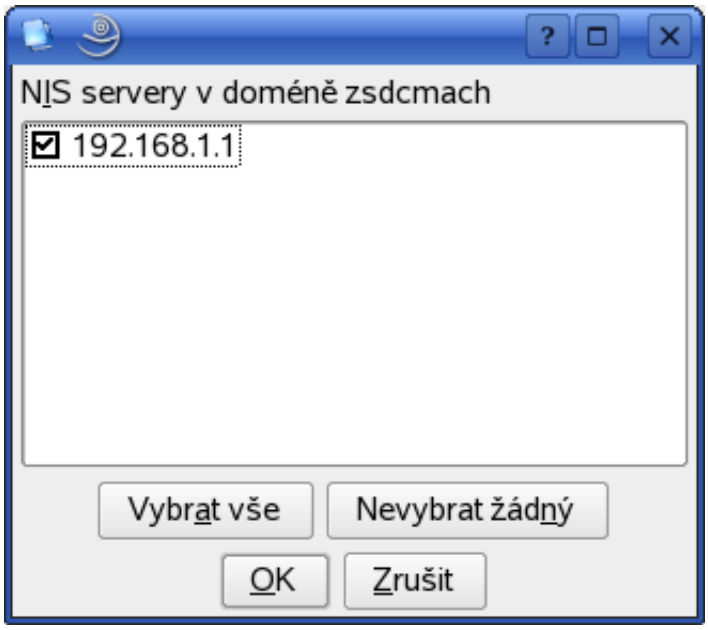
Současná nejnovější verze SuSE Linuxu má číslo 10.0 a je rozložena na pěti instalačních CD nebo dvou DVD. Její volně dostupnou variantu (tzv. OpenSuse) lze stáhnout například na adrese `ftp://ftp.linux.cz/pub/linux/suse/i386/10.0/iso/`[21]. Při instalaci se systém opět pokusí získat síťové informace ze serveru DHCP, a tak našimi jedinými úkoly bude připojení diskového oddílu `/home` ze serveru `orpheus`, případné připojení do domény NIS, popř. zapnutí autentizace proti službě LDAP a přidání síťové tiskárny.

Pro připojování vzdálených diskových oddílů je nutné mít na stanicích nainstalován balík `nfs-utils`. K připojování domovských adresářů uživatelů při startu stanic je možné využít příslušný ovládací modul YaST nebo upravit soubor `/etc/fstab` přidáním následujícího řádku místo původního, který definoval připojení oddílu `/home`:

```
192.168.1.1:/home /home nfs defaults 0 0
```

Jako zdrojový diskový oddíl je použit oddíl `/home` ze serveru `orpheus` (který má na svém vnitřním síťovém rozhraní nastavenu IP adresu 192.168.1.1), přípojným bodem na

[21] Za zmínku stojí také linuxové distribuce upravené přímo pro školní prostředí, například nová distribuce *Edubuntu*, obsahující množství výukových programů.



Obrázek 6: Připojení do domény NIS v SuSE Linuxu

stanici je adresář `/home`, jedná se o souborový systém `nfs` a parametrům pro připojení svazku jsou ponechány výchozí hodnoty. Je nutné vědět, že vlastnosti připojení oddílu podléhají nastavení exportovaného adresáře na serveru `orpheus`, viz. kapitola 10.1. Nyní již stačí počítač restartovat nebo spustit příkaz `mount -a -t nfs` pro připojení všech souborových systémů typu `nfs` uvedených v souboru `/etc/fstab`. Pokud bychom chtěli našim uživatelům vytvořit ještě podobnější prostředí, jaké mají v systému Windows, bude nutné stejným způsobem exportovat a připojovat také adresář `/tmp`, který se v prostředí Windows mapuje na síťovou jednotku (adresáře `install` a `vyuka` se nacházejí v oddílu `/home`, takže není nutné je explicitně exportovat, resp. připojovat).

Pokud je k autentizaci uživatelů vybrána síťová služba NIS, bude nutné stanice přidat do její domény. Na SuSE Linuxu se toto dá pohodlně udělat pomocí modulu *Klient NIS* ovládacího systému YaST. Ten v zadané doméně najde přístupné servery NIS a poté je možné k nim počítač připojit tak, jak je zobrazeno na obrázku 6. Po restartu počítače již budou na stanici dostupné informace o uživatelích serveru `orpheus`.

Byla-li k autentizaci uživatelů použita adresářová služba LDAP, v SuSE Linuxu lze její klientskou část opět velice jednoduše aktivovat buď již při instalaci systému nebo později pomocí modulu YaST, klient LDAP. Při této příležitosti může systém SuSE Linux požádat o instalační médium (či on-line zdroj), aby mohl nainstalovat všechny potřebné programy

a knihovny (`ldap-client, libnss-ldap, libpam-ldap`).

Pro urychlení práce na klientských počítačích je k přidání síťové tiskárny opět možné využít grafickou nástavbu. Díky aplikaci `smbclient` je i v Linuxu možné připojit sdílenou tiskárnu serveru Samba. Tím odpadá práce s nastavováním linuxového tiskového systému (například CUPS) pro sdílení tiskárny v síti včetně jeho zabezpečení a výsledek je naprosto dostačující. Navíc změna vlastností tiskového sdílení serveru Samba na serveru `orpheus` se tak dotkne obou operačních systémů na klientských počítačích.

16 Závěr

Daná diplomová práce se zabývala návrhem a popisem úseku počítačové sítě na Základní škole Máchovo náměstí Děčín. Cílem bylo vzít v úvahu všechny současné nedostatky sítě a další požadavky školy a navrhnout a popsat síťové řešení, které by všechny tyto nedostatky eliminovalo. Školní síť v tomto stavu řeší všechny současné nevýhody v učebně VT2 a tím pádem splňuje požadavky školy. Na žákovských počítačích je nainstalován jak systém Linux, tak i MS Windows. Škola tedy může nabídnout rozšíření běžného učiva výpočetní techniky. Zároveň je možné pracovat s mnoha výukovými programy určenými pro systém Windows. Uchovávání uživatelských jmen a hesel na serveru `orpheus` umožnujě přihlašování uživatelů z jakéhokoliv počítače v učebně VT2. Jejich synchronizace je zajištěna pomocí služeb NIS, `rsync + ssh` nebo adresářové služby LDAP. Sdílení domovských (i jiných) adresářů všech uživatelů zrychluje a zpřehledňuje výuku. Paketový firewall pomocí `iptables` zajišťuje zabezpečení vnitřní sítě a přístup na nevhodné internetové stránky je kontrolován a zabezpečován pomocí proxy serveru `squid`. Tiskárna v učebně VT2 je nyní sdílená a tudíž k dispozici všem počítačům v učebně, a to z obou operačních systémů. Konečně veškerá správa uživatelů, hesel, uživatelských práv, tisku, filtrace webového obsahu, IP adres i firewallu probíhá na jednom místě – na serveru `orpheus` (ať už lolálně nebo vzdáleným přístupem pomocí `ssh`). Tím je splněn i poslední požadavek školy – nutnost centralizované správy sítě.

Literatura

[1] DROMS, R., LEMON, T.: *DHCP – Příručka administrátora.* 1. vyd. Brno: Computer Press, 2004. 490 s. ISBN 80-251-0130-4.

[2] ECKSTEIN, R., COLLIER-BROWN, D., KELLY, P.: *Samba.* 1. vyd. Praha: Computer Press, 2001. 378 s. ISBN 80-7226-463-X.

[3] FUCHS, J.: *Bash IV* [online]. c2003, [cit. 23. 1. 2006]. `http://www.abclinuxu.cz/clanky/navody/bash-iv`.

[4] HORÁK, J, KERŠLÁGER, M.: *Počítačové sítě pro začínající správce.* 2. aktualizované vyd. Praha: Computer Press, 2003. 178 s. ISBN 80-7226-876-7.

[5] HUNT, C.: *Linux – Síťové servery.* 1. vyd. Praha: SoftPress, 2003. 672 s. ISBN 80-86497-59-3.

[6] KRČMÁŘ, P.: *Linux – Tipy a triky pro bezpečnost.* 1. vyd. Praha: Grada, 2004. 207 s. ISBN 80-247-0812-4.

[7] NEMETH, E., SNYDER, G., HEIN, T.R.: *Linux – Kompletní příručka administrátora.* 1. vyd. Brno: Computer Press, 2004. 828 s. ISBN 80-722-6919-4.

[8] *O projektu Indoš* [online]. 2003 [cit. 01. 12. 2005]. `http://www.indos.cz/oprojektu/`.

[9] OSTERLOH, H.: *TCP/IP – Kompletní průvodce.* 1. vyd. Praha: SoftPress, 2003. 512 s. ISBN 80-86497-34-8.

[10] PŘISPĚVATELÉ WIKIPEDIE, *Ethernet* [online], Wikipedie: Otevřená encyklopedie, c2005, Datum poslední revize 21. 12. 2005, 07:40 UTC, [cit. 16. 01. 2006] `<http://cs.wikipedia.org/w/index.php?title=Ethernet&oldid=285064>`.

[11] TS, J., ECKSTEIN, R., COLLIER-BROWN, D.: *Using Samba, second edition.* Second Edition. Boston, USA: O'Reilly Associates, 2003. 556 s. ISBN 0-596-00256-4.

[12] *What is Samba?* [online]. 2004. [cit. 01. 09. 2005]. `<http://us5.samba.org/samba/what_is_samba.html>`.

[13] WIKIPEDIA CONTRIBUTORS, *Netfilter/iptables* [online], Wikipedia, The Free Encyclopedia, c2006, Datum poslední revize 21. 1. 2006, 21:39 UTC, [cit. 25. 01. 2006] `http://en.wikipedia.org/wiki/Netfilter`.

Přílohy

A Výpis souboru mass-useradd.py

```
#!/usr/bin/env python
# skript pro hromadne pridani uzivatelu do databaze Linuxu
# pouziti:
##  python mass-useradd.py < vstup.txt >> nova-hesla.txt
#
#
from string import lowercase, uppercase
from random import choice
from os import system, popen
from sys import stdin, stdout


def make_pass():
    """vytvori nahodne, 8 mistne heslo ze znaku A-Za-z0-9."""
    numbers = "".join(map(str, range(10)))
    chars = lowercase + uppercase + numbers

    p = ""
    for r in range(8):
        p += choice(chars)

    return p

# Toto zpusobi expiraci hesla po prvnim prihlaseni
# a uzivatel bude muset zadat nove
#olddate = "2003-01-01"


# tato cast cte udaje ze vstupniho souboru
```

```
# a generuje vytvori nove heslo
# format souboru musi byt:
# login:Jmeno Prijmeni
for line in stdin.readlines():
    line = line.strip()  # odstrani znaky konce radku
    username, realname = line.split(':')
    password = make_pass()

# tato cast cte /etc/passwd a posouva UID + 1
    chp = popen("cat /etc/passwd | cut -f3 -d: | sort -un | tail -1 |
          awk '$id == $1 { $id++; print $id }'", 'r')
    next_id = int(chp.read())
    chp.close()

# uzivatele jsou pridani do /etc/passwd
# a prava domaciho adr nastavena na 700
    system("/usr/sbin/useradd -m -k /etc/skel -s /bin/bash -c \"%s\"
    -g 102 -u %d %s" % (realname, next_id, username))
    system("/bin/chmod 700 /home/%s" % (username))

# nastaveni hesla
    chp = popen("/usr/sbin/chpasswd", 'w')
    chp.write("%s:%s\n" % (username, password))
    chp.close()

# expirace hesla nastavenim posledni zmeny na 'davnou minulost'
# dalsi expirace za 90 dni - lze zmenit
#    system("/usr/bin/chage -M 90 -d %s %s" % (olddate, username))

# tisk informaci
    print "%s\t%s\t%d" % (username, password, next_id)
```

B Výpis souboru smb.conf

```
[global]
   workgroup = VT2
   netbios name = orpheus
   security = user
   server string = Samba Server
   encrypt passwords = yes
   hosts allow = 192.168.1.
   log file = /var/log/samba.%m
   max log size = 50
   passdb backend = tdbsam smbpasswd
   socket options = TCP_NODELAY
   interfaces = eth1
   local master = yes
   os level = 65
   domain master = yes
   preferred master = yes
   domain logons = yes
   logon script = logon.bat
   logon path = \\%L\Profiles\%U
   wins support = yes
   dns proxy = no
   add user script = /usr/sbin/useradd %u
   add group script = /usr/sbin/groupadd %g
   add machine script = /usr/sbin/useradd -c "Machine account"
    -d /dev/null -g 105 -s /bin/false %u
   delete user script = /usr/sbin/userdel %u
   delete user from group script = /usr/sbin/deluser %u %g
   delete group script = /usr/sbin/groupdel %
   hide dot files = yes
   time server = yes
   guest ok = no
   logon drive = H:
```

```
   null passwords = no
   passwd program = /usr/bin/passwd %u
   unix password sync = yes
   load printers = yes
   printing = cups

[homes]
   comment = Domovsky adresar
   browseable = no
   guest ok = no
   writable = yes

[profiles]
   path = /home/roaming
   browsable = no
   writable = yes
   create mask = 0600
   directory mask = 0700

[netlogon]
   comment = Network Logon Service
   path = /etc/samba/.winlogon
   guest ok = no
   writable = no
   browsable = no

[vyuka]
   comment = Soubory pro vyuku
   path = /home/share/vyuka
   public = no
   printable = no
   write list = @ucitele
   force group = ucitele
```

```
[verejne]
   comment = Verejne sdileni
   path = /tmp
   read only = no
   public = no

[install]
   comment = Instalacni soubory
   path = /home/share/install
   read only = yes
   public = no

[printers]
   comment = Tiskarny na serveru
   path = /var/spool/samba
   guest ok = no
   writable = no
   printable = yes
```

C Výpis souboru addsmb.sh

```
#!/bin/bash
# addsmb.sh
# program pro hromadne pridavani uzivatelu do databeze Samby
#
# pouziti: ./addsmb.sh soubor_se_jmeny_a_hesly vystup_pro_kontrolu
if [ "$1" == "" ]; then
        echo "Je nutne zadat soubor se jmeny a hesly jako argument!";
        exit;
fi
if [ "$2" == "" ]; then
        echo "Nebyl zadan soubor s pozadovanym vystupem!";
        echo "Jestli je to tak spravne, pockejte 5 vterin ...";
        echo "Muzete stisknout Ctrl+C a program se ukonci";
        sleep 5;
fi
FILE="$1";
cat $FILE | while read line;
do
        name=$(echo ${line} | awk '{ print $1}');
        password=$(echo ${line} | awk '{ print $2}');
        (echo $password; echo $password) | smbpasswd -s -a $name
        if [ "$2" != "" ]; then
                echo "$name     $password" >> $2
        fi
done
#
# konec programu
```

D Výpis souboru syncpwd.sh

```
#!/bin/bash
#
# Program pro synchronizaci souboru pomoc rsync a ssh
#
##
PREFIX="192.168.1"      # sit
FIRST=10                # prvni klient v siti
LAST=34                 # posledni klient v siti
FILES="/etc/passwd /etc/shadow /etc/group" # soubory
LOGFILE=syncpwd.log     # kam logovat neuspesny sync
##
#
while  [ ! $FIRST -gt $LAST ]
do
     if ping -c1 $PREFIX.$FIRST > /dev/null 2>&1
     then
        if rsync -p -e ssh $FILES $PREFIX.$FIRST:/etc 2>> $LOGFILE
        then
           echo "Sync s $PREFIX.$FIRST byl uspesny."
        else
           echo "Sync s $PREFIX.$FIRST neuspesny!!"
        fi
        else
           echo "Host $PREFIX.$FIRST je vypnuty."
     fi
     let FIRST=FIRST+1
done
```

E Výpis souboru newuser.sh

```
#!/bin/bash
#
# Program newuser.sh:
# 0. Ziska informace o novem uzivateli
# 1. Prida noveho uzivatele do unixove databaze
# 2. Prida noveho uzivatele do databaze Samby
# 3. Nastavi mu souborove kvoty podle vytvorenych prototypu
# 4. Aktualizuje databazi NIS
#
# K.Kantar
#
# config:
PWDFILE="/root/.hesla$$"
HOMEDIR=/home
NISDBDIR=/var/yp
#
#
if [ `id -u` -ne 0 ]; then
        echo "Musite byt root, jinak program nebude fungovat"
        exit
fi
echo
# ************************************************************
echo "*** Pridani noveho uzivatele ***"
echo -n "* Zadej prihlasovaci jmeno: "
CONT=""
until [ "$CONT" != "" ]; do
        read LOGIN
        until [ "$LOGIN" != ""  ]; do
                echo "Musite zadat prihlasovaci jmeno!"
                echo -n "* Zadej prihlasovaci jmeno: "
                read LOGIN
```

```
        done
        EXIST=‘cat /etc/passwd | awk -F":" ’{ print $1 }’ |
        grep -w $LOGIN‘
        if [ "$EXIST" == "" ]; then
                CONT="OK"
        else
                CONT=""
                echo "Toto jmeno jiz existuje! Vyberte jine."
                echo -n "* Zadej prihlasovaci jmeno: "
        fi
done
# ************************************************************
echo -n "* Zadej cele jmeno uzivatele: "
read NAME
until [ "$NAME" != ""  ]; do
        echo "Musite zadat cele jmeno!"
        echo -n "* Zadej cele jmeno uzivatele: "
        read NAME
done
# ************************************************************
PASSWD1=$$.$$‘date +%s‘
PASSWD2=$$‘date +%s‘

until [ $PASSWD1 == $PASSWD2 ]; do
        stty -echo
        read -p "* Zadej heslo: " PASSWD1; echo
        stty echo

        stty -echo
        read -p "* Zadej heslo znovu: " PASSWD2; echo
        stty echo

        if  [ $PASSWD1 != $PASSWD2 ]; then
```

```
                echo "Hesla nejsou stejna!"
        fi
done
# ************************************************************
GROUP=$$.$$‘date +%s‘
until [ "$GROUP" == "" ]; do
        echo -n "* Zadej skupinu: "
        read SKUPINA
        GROUP=‘cat /etc/group | awk -F":" ’{ print $1 }’ |
        grep -w $SKUPINA‘
        if [ "$GROUP" != "" ]; then
                GROUP=""
        else
                GROUP="OK"
                echo "Tato skupina neexistuje, zadejte jinou!"
fi
done
# ************************************************************
echo
echo
echo "+++ Zadane prihlasovaci jmeno: $LOGIN"
echo "+++ Zadane cele jmeno: $NAME"
echo "+++ Zadana skupina: $SKUPINA"
echo
echo -n "Zpracovavam uzivatele..."
sleep 2
echo
echo

echo "$LOGIN:$PASSWD1" > $PWDFILE
useradd -m -k /etc/skel -s /bin/bash -c "$NAME"
-d $HOMEDIR/$LOGIN -g $SKUPINA $LOGIN \
&& echo "** Uzivatel $LOGIN (uid ‘cat /etc/passwd | grep -w $LOGIN |
```

```
awk -F: '{ print $3 }'`) pridan do unixove databaze **"

chpasswd < $PWDFILE
rm -f $PWDFILE # lze zakomentovat, pro zachovani souboru
chmod 700 /home/$LOGIN \
&& echo "** Domovsky adresar $HOMEDIR/$LOGIN byl nastaven **"

(echo $PASSWD1; echo $PASSWD1) | smbpasswd -s -a $LOGIN > /dev/nul \
&& echo "** Uzivatel $LOGIN pridan do databeze Samby **"

edquota -p prototyp_$SKUPINA $LOGIN \
&& echo "** Diskova kvota pro uzivatele $LOGIN byla nastavena **"

cd $NISDBDIR && make > /dev/null 2>&1 \
&& echo "** Databaze NIS byla aktualizovana **"
```

F Výpis souboru firewall.sh

```
#!/bin/sh
#
# K.Kantar
#
# Vnejsi IP a rozhrani
INET_IP="172.16.1.31" # pouze priklad
INET_IFACE="eth0"

# Vnitrni adresa, broadcast a rozhrani
LAN1_IP="192.168.1.1/24"
LAN1_BCAST="192.168.1.255/24"
LAN1_IFACE="eth1"

# Loopback rozhrani
LO_IFACE="lo"
LO_IP="127.0.0.1/32"

# Cesta k iptables
IPTABLES="/sbin/iptables"

##################################################

# Inicializace modulu
/sbin/depmod -a

# Zavedeme modulu
/sbin/modprobe ipt_LOG
/sbin/modprobe ipt_REJECT
/sbin/modprobe ipt_MASQUERADE

# Zapneme routovani
echo "1" > /proc/sys/net/ipv4/ip_forward
```

```
echo "1" > /proc/sys/net/ipv4/tcp_syncookies

# rp_filter na zamezeni falesnych IP
for interface in /proc/sys/net/ipv4/conf/*/rp_filter; do
   echo "1" > ${interface}
done

# Pro jistotu vymaz existujicich pravidel
$IPTABLES -X
$IPTABLES -F

# Implicitni politika
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP


####################################
# Retezec PREROUTING v NAT tabulce #
####################################

# Presmerovani vsech pozadavku na port 80 na proxy (port 3128)
$IPTABLES -t nat -A PREROUTING -i $LAN_IFACE -p tcp --dport 80 -j \
REDIRECT --to-port 3128

# Zahazovani dotazu z rezervovanych adres
$IPTABLES -t nat -A PREROUTING -i $INET_IFACE -s 192.168.0.0/16 -j DROP
$IPTABLES -t nat -A PREROUTING -i $INET_IFACE -s 172.16.0.0/12 -j DROP
$IPTABLES -t nat -A PREROUTING -i $INET_IFACE -s 10.0.0.0/8 -j DROP
$IPTABLES -t nat -A PREROUTING -i $INET_IFACE -s 127.0.0.0/8 -j DROP
$IPTABLES -t nat -A PREROUTING -i $INET_IFACE -s 224.0.0.0/4 -j DROP
$IPTABLES -t nat -A PREROUTING -i $INET_IFACE -s 96.0.0.0/4 -j DROP
```

```
#####################################
# Retezec POSTROUTING v NAT tabulce #
#####################################

# IP maskarada - SNAT = konektivita vnitrni site
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -j SNAT --to $INET_IP


###################
# Retezec FORWARD #
###################

# Paket navazuje spojeni, ale nema nastaveny priznak SYN
$IPTABLES -A FORWARD -p tcp ! --syn -m state --state NEW -j DROP
$IPTABLES -A FORWARD -p tcp -i $INET_IFACE --tcp-flags SYN,FIN SYN,FIN \
-j DROP

# Routing zevnitr site ven
$IPTABLES -A FORWARD -i $LAN1_IFACE -j ACCEPT

# Stavovy firewall
$IPTABLES -A FORWARD -i $INET_IFACE -o $LAN1_IFACE -m state --state \
ESTABLISHED,RELATED -j ACCEPT

#################
# Retezec INPUT #
#################

# Paket navazuje spojeni, ale nema nastaveny priznak SYN
$IPTABLES -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

# Portscan
$IPTABLES -A INPUT -p tcp -i $INET_IFACE --tcp-flags SYN,FIN SYN,FIN \
```

```
-j DROP

# Propoustime ICMP dotazy - nektere
$IPTABLES -A INPUT -i $INET_IFACE -p ICMP --icmp-type 0 -j ACCEPT
$IPTABLES -A INPUT -i $INET_IFACE -p ICMP --icmp-type 3 -j ACCEPT
$IPTABLES -A INPUT -i $INET_IFACE -p ICMP --icmp-type 8 -j ACCEPT
$IPTABLES -A INPUT -i $INET_IFACE -p ICMP --icmp-type 11 -j ACCEPT

# Loopback
$IPTABLES -A INPUT -i $LO_IFACE -j ACCEPT

# Pakety z lokalni site pro nas
$IPTABLES -A INPUT -i $LAN1_IFACE -d $LAN1_IP -j ACCEPT
$IPTABLES -A INPUT -i $LAN1_IFACE -d $INET_IP -j ACCEPT

# Broadcasty na lokalnim rozhran
$IPTABLES -A INPUT -i $LAN1_IFACE -d $LAN1_BCAST -j ACCEPT

# MS klienti maji chybu v implementaci DHCP
$IPTABLES -A INPUT -i $LAN1_IFACE -p udp --dport 67 -j ACCEPT

# Pakety od navazanych spojeni
$IPTABLES -A INPUT -d $INET_IP -m state --state ESTABLISHED,RELATED \
-j ACCEPT

###################
# Retezec OUTPUT #
###################

# Povolime odchozi pakety s nasimi IP
$IPTABLES -A OUTPUT -s $LO_IP -j ACCEPT
$IPTABLES -A OUTPUT -s $LAN1_IP -j ACCEPT
$IPTABLES -A OUTPUT -s $INET_IP -j ACCEPT
```

```
# Povolime vnitrni DHCP broadcasty
$IPTABLES -A OUTPUT -o $LAN1_IFACE -p UDP --dport 68 --sport 67 \
-j ACCEPT
```

G Obsah CD-ROM

Součástí této diplomové práce je CD-ROM obsahující její elektronickou verzi ve formátech PDF a PostScript, její zdrojový kód ve formátu LaTeX a všechny zmíněné programy a skripty. Disk CD-ROM obsahuje následující adresáře:

- `thesis-source` – obsahuje zdrojové kódy práce ve formátu LaTeX a všechny obrázky ve formátech PNG a JPG.
- `thesis` – obsahuje přeložený dokument ve formátech PDF a PostScript.
- `scripts` – obsahuje všechny skripty popisované v diplomové práci, které zároveň tvoří její přílohy A – F.