

Boot GNU/Linuxu pomocí PXE a provoz s terminal serverem

Boot of GNU/Linux using PXE and operation with terminal server

Tomáš Bezděk

Bakalářská práce
2007



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2006/2007

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš BEZDĚK**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Boot GNU/Linuxu pomocí PXE a provoz s terminal serverem.**

Zásady pro vypracování:

Vypracujte literární řešení na dané téma.

Nakonfigurujte bezdiskovou stanici tak, aby systém nabootoval pomocí PXE.

Nakonfigurujte spolupraci s terminal serverem.

Podrobně popište postup a možnosti konfigurace.

Předvedte reálné nasazení.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

Deitel, H. M.: *Operating Systems*, Prentice Hall, 2004

Tanenbaum, A. S.: *Modern operating systems*, Prentice Hall, 2002

Linux – Dokumentační projekt, Computer Press, 2003

Sobell, M., G.: *Linux-praktický průvodce*, ComputerPress, 1999.

Nemeth, E., Snyder, G., Hein, T. R.: *Linux – kompletní příručka administrátora*. ComputerPress, 2004.

Vedoucí bakalářské práce:

Ing. Martin Sysel, Ph.D.

Ústav aplikované informatiky

Datum zadání bakalářské práce:

13. února 2007

Termín odevzdání bakalářské práce:

24. května 2007

Ve Zlíně dne 13. února 2007



prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Bakalářská práce je zaměřena na zavedení operačního systému Linux ze sítě za pomoci prostředí PXE a na vytvoření grafického terminálu k terminálovému serveru. Celá práce se skládá ze dvou částí.

V teoretické části čtenář získá informace o možnostech využití zavedení systému ze sítě, použitých protokolech a principu zavedení systému ze sítě.

Praktická část je zaměřena na vytvoření terminálové sítě s ohledem zejména na využití pro UTB Zlín. Čtenář zde může získat cenné informace pro výstavbu terminálové sítě a zároveň zde najde návod na instalaci kompletní terminálové sítě. Celá instalace je rozdělena na části podle toho, o kterou komponentu terminálové sítě se jedná.

Klíčová slova: Linux, PXE, Terminál, DHCP, NFS, TFTP, XDM, X Window System

ABSTRACT

This bachelor work is aimed to boot operating system Linux over network using PXE and to build graphical terminal to terminal server. The work is divided in to two parts.

In the theoretical part reader acquires information's about possibilities of usage the network boot and about used protocols.

Practical part is aimed to the build of the terminal network in relation to usage at UTB in Zlín. Reader can acquire information's to build the terminal network and can find installation guide for complete terminal network. All installations are divided by components of terminal network.

Keywords: Linux, PXE, Terminal, DHCP, NFS, TFTP, XDM, X Window System

Tímto chci poděkovat vedoucímu bakalářské práce panu Ing. Martinu Syslovi, Ph.D. za odborné vedení, cenné rady a připomínky, které mi poskytoval při řešení této práce.

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně, dne 24.5.2007

.....
Podpis diplomanta

OBSAH

| | |
|--|-----------|
| ÚVOD | 8 |
| I TEORETICKÁ ČÁST | 9 |
| 1 VÝSTAVBA TERMINÁLOVÉ SÍTĚ S VYUŽITÍM OPERAČNÍHO SYSTÉMU LINUX | 10 |
| 1.1 NÁVRH MODERNÍ TERMINÁLOVÉ SÍTĚ | 10 |
| 1.2 LINUX..... | 11 |
| 1.3 PXE - PREBOOT EXECUTION ENVIRONMENT..... | 12 |
| 1.4 X WINDOW SYSTEM..... | 12 |
| 1.5 XDM - X DISPLAY MANAGER | 14 |
| 1.6 DHCP - DYNAMIC HOST CONFIGURATION PROTOCOL..... | 15 |
| 1.7 TFTP – TRIVIAL FILE TRANSFER PROTOCOL | 15 |
| 1.8 NFS – NETWORK FILE SYSTEM..... | 16 |
| 1.9 MOŽNOSTI VYUŽITÍ ZAVEDENÍ OPERAČNÍHO SYSTÉMU ZE SÍTĚ | 16 |
| 1.9.1 Bezdiskový klient, server nebo uzel výpočetního clusteru..... | 16 |
| 1.9.2 Terminál k terminálovému serveru | 16 |
| 1.9.3 Instalace operačního systému ze serveru | 17 |
| II PRAKTICKÁ ČÁST | 18 |
| 2 NASAZENÍ TERMINÁLOVÉ SÍTĚ | 19 |
| 2.1 VÝBĚR POUŽÍVANÉHO HARDWARE | 21 |
| 2.2 VÝBĚR POUŽÍVANÉHO SOFTWARE | 21 |
| 2.3 PRAKTICKÝ START LINUXU ZE SÍTĚ S VYUŽITÍM PXE..... | 22 |
| 2.4 POUŽITÝ HARDWARE..... | 23 |
| 2.5 POUŽITÉ SÍŤOVÉ PRVKY..... | 23 |
| 3 NÁVOD NA INSTALACI SERVERŮ A SESTAVENÍ SÍTĚ TERMINÁLŮ | 24 |
| 3.1 PROPOJENÍ TERMINÁLOVÉHO SERVERU, BOOT SERVERU A TERMINÁLU DO SÍTĚ..... | 24 |
| 3.2 INSTALACE A NASTAVENÍ TERMINÁLOVÉHO SERVERU..... | 24 |
| 3.3 INSTALACE A NASTAVENÍ BOOT SERVERU | 26 |
| 3.4 VYTVOŘENÍ SYSTÉMU PROVOZOVANÉHO NA BEZDISKOVÝCH TERMINÁLECH | 30 |
| 3.4.1 Vytvoření části systému určené pro provoz na terminálu..... | 31 |
| 3.4.2 Vytvoření části připojované z boot serveru | 35 |
| ZÁVĚR | 38 |
| ZÁVĚR V ANGLIČTINĚ | 39 |
| SEZNAM POUŽITÉ LITERATURY | 40 |
| SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK | 41 |

| | |
|----------------------|----|
| SEZNAM OBRÁZKŮ | 42 |
| SEZNAM PŘÍLOH..... | 43 |

ÚVOD

S rozvojem využití počítačů a počítačových sítí se objevují snahy zefektivnit provoz většího množství počítačů nahrazením konvenčního vybavení nebo změnou modelu provozu. Požadavkem mohou být úspory nákladů na údržbu a provoz nebo požadavky na vyšší bezpečnost. Jedním z vhodných způsobů, jak splnit většinu těchto požadavků je využití terminálové sítě.

Terminál oproti plnohodnotnému počítači neobsahuje některé části, které se pochopitelně také projevují zejména na ceně a energetických nárocích takového zařízení.

V současné době je většina terminálů na trhu založena na architektuře počítačů třídy PC. Je rovněž možné sestavit terminál z komponent pro běžné stolní počítače. Při využití komponent pro stolní počítače je oproti běžnému počítači vynechán zejména pevný disk. Pro zavedení operačního systému terminálu lze s úspěchem využít prostředků sítě, do níž je terminál připojen. Jedním z prostředků terminálu pro zavedení systému ze sítě je PXE. Vhodný operační systém pro provoz celé terminálové sítě je například Linux.

O nasazení terminálové sítě je uvažováno ve studovnách a učebnách na Fakultě aplikované informatiky UTB ve Zlíně a ve studovnách na kolejích UTB ve Zlíně.

Ve 3. kapitole se nachází návod na instalaci kompletní terminálové sítě obsahující instrukce provádějící instalaci krok za krokem.

I. TEORETICKÁ ČÁST

1 VÝSTAVBA TERMINÁLOVÉ SÍTĚ S VYUŽITÍM OPERAČNÍHO SYSTÉMU LINUX

Pod pojmem terminálová síť rozumíme síť, v níž úlohu pracovních stanic zajišťují terminály připojené k serveru, a to tak, že veškeré operace s daty a výpočty probíhají na terminálovém serveru. Tento přístup se objevil v 60. letech 20. století za účelem zpřístupnění počítačů uživatelům, protože počítače tehdy byly velmi drahé a prostorově náročné. Terminály tehdy byly výhradně znakové a se serverem byly propojeny sériovou linkou.

Pojem terminál znamená zařízení, které slouží k přístupu uživatele k terminálovému serveru a slouží pouze k zobrazování výstupů aplikací a k odesílání uživatelských vstupů na server.

V současné době dochází opět k nasazování terminálů namísto plnohodnotných stanic zejména z důvodů úpory energií a nákladů na správu hardware i software. Moderní terminály jsou výhradně grafické a bývají s terminálovým serverem propojeny zejména pomocí sítě typu ethernet.

Jako terminál je možné využít přímo specializované zařízení (například: Fujitsu Siemens Futro nebo HP ThinClient), optimalizované výrobcem pro využití jako terminál. Tato zařízení bývají obvykle vybaveny úspornými komponenty a neobsahují nic navíc, než co vyžaduje funkce terminálu.

Další variantou je využití běžného kancelářského počítače, který by byl ochuzen o součásti, které funkce terminálu nevyžaduje.

Jako terminálový server je možné využít jakýkoliv počítač nebo server, případně cluster, který umožňuje provoz použitého operačního systému a propojení s terminály.

Jako operační systém pro provoz terminálů a serverů v terminálové síti byl zvolen operační systém Linux zejména pro svou cenu, příznivou licenční politiku a možnost jednoduše v systému provádět jakékoliv změny.

1.1 Návrh moderní terminálové sítě

Terminálovou síť lze rozdělit na 3 základní části: síťová infrastruktura, servery a uživatelské terminály. U síťové infrastruktury se jedná o kabelové rozvody a aktivní

síťové prvky. U uživatelů bude instalován vlastní terminál, klávesnice, myš a displej nebo obrazovka.

Servery budou instalovány v dedikovaných prostorách spolu s aktivními prvky síťové infrastruktury. Servery budou zastávat 2 základní úlohy: start a provoz systému na terminálu a provoz uživatelských aplikací. U menších instalací může obě úlohy zastávat jeden server.

Start a provoz systému na terminálu budou obstarávat tzv. boot servery. Zavedení systému pomocí PXE vyžaduje služby jako DHCP a TFTP jejichž provozování v rozlehlých sítích (například dvě vzdálené pobočky) je prakticky nerealizovatelné. Možným řešením je instalovat v každé pobočce jeden boot server a například v centrální pobočce jeden terminálový server.

Uživatelské aplikace budou provozovány na terminálovém serveru. Terminálový server je vhodné vybavit více procesory, popřípadě vybudovat cluster z více serverů.

1.2 Linux

Linux je jádro počítačových operačních systémů vyvíjené pod licencí GNU/GPL. Jeho vývoj začal v roce 1991 jako koníček finského studenta Linuse Torvaldse, později se do vývoje zapojilo větší množství nadšenců. V současné době vývojem do jádra přispívají velké společnosti jako RedHat Inc. Systém rovněž získává podporu na serverech od velkých společností jako IBM, Hewlett-Packard a Novell.

Původní vývoj byl zaměřen na 32 bitové procesory 386 a kompatibilní. V průběhu času byla zařazena podpora většiny 32 a 64 bitových architektur. Je tedy možné jeho nasazení v široké škále zařízení od mobilních telefonů až po superpočítače.

Protože Linux je pouze jádro operačního systému, je pro jeho použití jádro sdružit do distribuce s dalšími nástroji, knihovnami a programy. Většina použitelných nástrojů (bash, emacs, gcc, glib) byla vytvořena v rámci projektu GNU krátce před vytvořením jádra Linux. V distribucích bývá mimo aplikace uveřejněné pod licencí GNU/GPL jádro Linux a nástroje z projektu GNU, proto distribuce bývají obvykle označovány jako GNU/Linux například: Debian GNU/Linux. Distribuce, ve kterých jsou programy, nástroje nebo ovladače které nejsou dostupné pod licencí GNU/GPL, nebývají označovány GNU/Linux.

Takto tomu je například u distribucí RedHat Enterprise Linux a SUSE Linux Enterprise. [6]

1.3 PXE - Preboot Execution Environment

PXE je prostředí umožňující zavést operační systém počítače s využitím síťového rozhraní nezávisle na datových úložištích nebo instalovaném operačním systému.

Tento standard byl společností Intel představen jako část Wired for Management framework verze 2.1. PXE využívá standardních síťových protokolů IP,UDP, DHCP a TFTP pro rozšíření běžného firmware klientského síťového adaptéru.

Při startu PC a po skončení POST se firmware pokusí nalézt PXE službu pro přesměrování (DHCP) s informací o dostupných PXE boot serverech. Po zpracování odpovědi se firmware pokusí dotázat příslušného boot serveru na cestu k síťovému zaváděcímu programu. Stažení síťového zaváděcího programu do RAM počítače probíhá za pomoci TFTP, po stažení následuje případná kontrola chyb a vlastní spuštění síťového zaváděcího programu.

PXE byl navržen s ohledem na použití na mnoha různých systémových architekturách. Nicméně kompletní specifikace pokrývá pouze architekturu IA-32. Intel také PXE zahrnul do Extensible Firmware Interface pro svou architekturu IA-64.

V současné době je PXE dostupné prakticky na všech základních deskách s integrovaným síťovým rozhraním, případně stávající síťový adaptér je možné o PXE vybavit pomocí rozšiřující ROM.

V případě nemožnosti rozšířit síťový adaptér o PXE je možné využít startu PXE z CD nebo diskety. Pro start PXE z diskety je vhodný projekt etherboot (www.etherboot.org). [7]

1.4 X Window System

X Window System vznikl na Massachusetts Institute of Technology v roce 1984. Současná verze protokolu X11 vznikla v roce 1987. V současné době je leaderem v této oblasti X.Org Foundation.

Umožňuje vytvoření grafického uživatelského rozhraní (GUI) nejen na Unixových operačních systémech ale i na mnoha dalších moderních operačních systémech.

Pro vytvoření GUI poskytuje základní sadu nástrojů jako: vykreslování a přesouvání oken po obrazovce a jejich interakce s myší a klávesnicí.

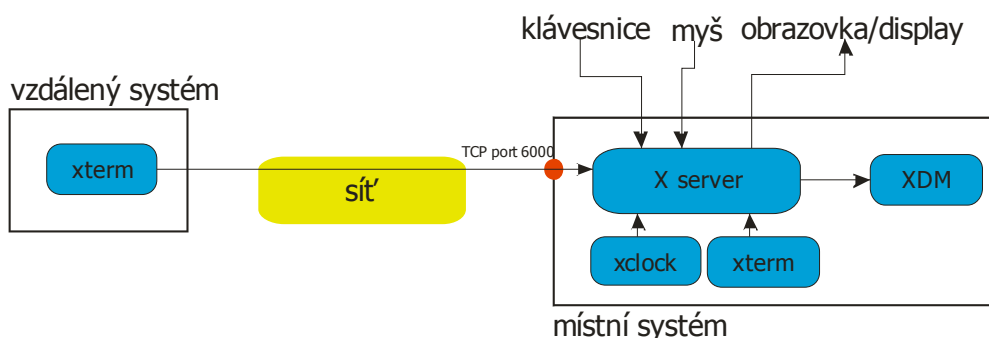
X samotné nespravuje uživatelské rozhraní, individuální klientská aplikace si své uživatelské rozhraní spravuje sama. Tento přístup umožňuje velkou variabilitu vzhledu a způsobu ovládání aplikací využívajících X.

Jednou ze základních vlastností X je síťová transparentnost, tím rozumíme, že program (klientská aplikace) může běžet na jiném počítači, než na kterém je zobrazováno její uživatelské rozhraní. Můžeme se setkat s termíny X-server a X-klient, kdy X-server běží na počítači na straně uživatele a X-klient je aplikace komunikující s X-serverem (X-server zobrazuje její uživatelské rozhraní).

Nevýhodou znemožňující nasazení v prostředí internetu nebo rozsáhlých sítí je absence jakéhokoliv šifrování komunikace mezi X-serverem a X-klientem. Tento problém může být řešen například tunelováním pomocí OpenVPN nebo IPsec. Jistým omezením může také být, že komunikace X-serveru s klientem nemůže být přerušena a znovunavázána v místě přerušení tak, jak tomu je například u VNC.

Design X Window System ve srovnání s grafickým subsystémem integrovaným v jádru operačního systému může vykazovat větší nároky na systémové prostředky počítače.

Na Obr. 1 je znázorněna funkce X Window systému například na běžném stolním počítači a připojení jedné aplikace skrze síť. Na místním systému je spuštěn kompletní X server, XDM a sada aplikací. Aplikace komunikují s X serverem, který zajišťuje vykreslování a zpracování vstupů například z klávesnice a myši. X server pro autorizaci uživatele využívá XDM serveru. Na vzdáleném systému je spuštěna aplikace, jejíž výstup je přeměřován přes síť na místní systém. Z pohledu uživatele se okno aplikace ze vzdáleného systému chová, jakoby se jednalo o místní aplikaci. [8]



Obr. 1: Princip funkce X serveru v lokálním a síťovém prostředí

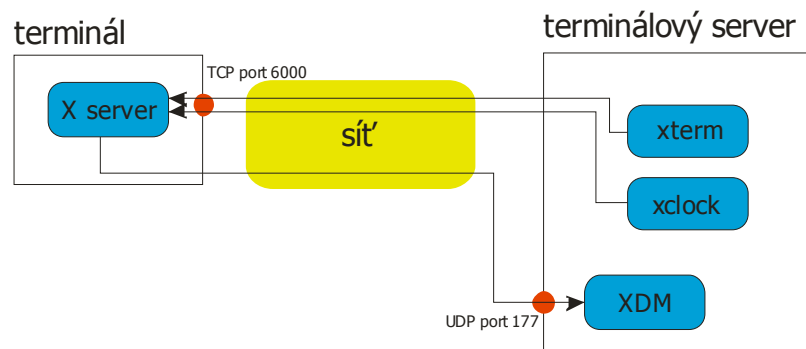
1.5 XDM - X Display Manager

Jde o součást systému X Window System, které zajišťuje vytvoření a správu sezení na X serveru ze stejného nebo vzdáleného počítače. X server na terminálu komunikuje s terminálovým serverem mimo jiné pomocí XDMCP (X Display Manager Control Protocol). Správce displeje XDM se uživateli prezentuje žádostí o uživatelské jméno a heslo. Sezení je aktivováno zadáním platného jména a hesla, jedná se tedy o úspěšné přihlášení k systému. Funkce systému v případě běhu na místním počítači je obdobná, jako funkce programů `init`, `getty` a `login` v textovém režimu. V případě, že správce displeje běží na vzdáleném počítači, pak se chová podobně jako `ssh` nebo `telnet` server, kdy si vyžádá jméno a heslo a případně spustí vzdálené sezení.

XDMCP využívá ke své síťové komunikaci UDP port 177. Komunikaci zahajuje X server, a to buď přímým dotazem na XDM server a nebo všesměrovým vysíláním, na které XDM server odpoví.

Nevýhodou XDMCP zůstává, že veškerá komunikace probíhá v nešifrované podobě. Pro provoz na nedůvěryhodné síti je proto nutné tento provoz šifrovat například pomocí tunelování přes virtuální síťové rozhraní.

Na Obr. 2 je znázorněna návaznost mezi X serverem na terminálu a terminálovým serverem. Aplikace na terminálovém serveru komunikují s terminálem skrze TCP port 6000 na terminálu. X server na terminálu komunikuje s XDM na terminálovém serveru skrze UDP port 177. [9]



Obr. 2: Princip funkce terminálu a terminálového serveru

1.6 DHCP - Dynamic Host Configuration Protocol

Je to sada pravidel používaných pro komunikaci zařízení, jako počítač, router nebo síťový adaptér za účelem vyžádání IP adresy ze serveru, který má seznam adres dostupných k přidělení.

Mimo IP adresu bývá také přidělována síťová maska, výchozí brána a adresy DNS serverů. DHCP server je vhodný pro rozlehlé sítě, kde by bylo manuální přidělování adres náročné pro správce sítě. Pro změnu adres stanic v síti stačí pouze změnit přidělovaný adresní prostor na serveru a aby stanice požádala o znovu přidělení adresy. Rovněž je zajištěna unikátnost přidělené IP adresy. DHCP server si uchovává, které IP adresy byly přiděleny a které jsou volné.

Nastavením je možné omezovat přidělování IP adres podle MAC adres, například jedné MAC adrese přidělovat pevnou IP nebo přidělovat IP adresy jen povoleným adresám. [10]

1.7 TFTP – Trivial File Transfer Protocol

Jde o velmi jednoduchý protokol pro přenos souborů, jehož funkce je velmi jednoduchou formou FTP protokolu. Pro svou jednoduchost je velmi nenáročný zejména na paměť a tím vhodný pro jednoduchá zařízení.

Bývá využíván jako prostředek pro zavedení zařízení, která nedisponují datovým úložištěm, které by obsahovalo operační systém nebo jiný software pro provoz zařízení. Protokol TFTP využívá ke komunikaci UDP port 69 narušil od TCP 21 u FTP. Bývá využíván ke stažení nebo nahrání souboru ve třech režimech: netascii, octet a mail. Režim netascii a octet odpovídají režimům ascii a bin (image) režimům u FTP, režim mail dnes už prakticky nebývá využíván. Vzhledem ke své jednoduchosti má TFTP také jistá omezení. Neumožňuje výpis obsahu adresáře a neobsahuje žádné autentizační a šifrovací mechanismy. Rovněž limit velikosti souboru je 32 MiB, který byl později rozšířen na 4 GiB. Protože TFTP na UDP supluje vlastnosti TCP spojení tak, že po vyslání balíku dat následuje potvrzení přijetí, není přenos dat nijak závratně rychlý i na linkách s vysokou propustností.

Pro bezpečnostní nedostatky je nevhodné využívat TFTP v rámci internetu. [11]

1.8 NFS – Network File System

Je síťový protokol pro vzdálený přístup k souborům přes síť. Na vývoji pracovala v roce 1984 společnost Sun Microsystems, v současnosti se o další vývoj stará organizace IETF.

NFS pracuje především nad transportním protokolem UDP, od verze 3 byla implementována podpora TCP.

V prostředí Linuxu lze protokol využít k připojení svazku ze vzdáleného serveru a pracovat s ním jako s lokálním. [12]

1.9 Možnosti využití zavedení operačního systému ze sítě

Mezi jednotlivými přístupy je několik zásadních rozdílů zejména co se týká rozdělení úloh mezi server a stanici, nebo vhodnosti jednotlivých přístupů pro různé aplikace.

1.9.1 Bezdiskový klient, server nebo uzel výpočetního clusteru

Tenký klient nebo server je síťový počítač, který oproti běžnému PC neobsahuje například pevný disk a jednotky pro vyměnitelná média.

Jedná se o model kdy veškeré programy, data ale i operační systém jsou uložena na serveru, z něhož je svazek připojen ke stanici například pomocí síťových protokolů NFS nebo SMB.

Výhodou tohoto přístupu je možné provozovat bezdiskové stanice nebo servery, kdy aplikace využívá systémové prostředky (vyjma úložného prostoru) bezdiskové stanice nebo serveru, ovšem se spouští ze serveru.

Nevýhodou je, že veškerá data a programy jsou neustále přesouvány sítí a v případě výpadku sítě nebo serveru s daty jsou všechny bezdiskové stanice nefunkční.

1.9.2 Terminál k terminálovému serveru

Tenký klient je síťový počítač, který bývá navržen podobně jako bezdiskový klient, ovšem se jedná o ještě více minimalisticky navrženou stanici (méně výkonný CPU, méně RAM, jednoduchý BIOS). Obvykle se nejedná o nijak výkonný počítač. Zpravidla bývá takovýto počítač navrhován s ohledem na cenu, velikost a spotřebu elektrické energie.

Jedná se o server-centrický model provozu stanic, kdy veškeré operace s daty, přístup k datům, přístup k paměťovému prostoru atd. řídí terminálový server případně aplikační server.

Typickým případem využití může být společnost nebo instituce, která potřebuje využívat větší množství stanic určených pro jeden účel, může se jednat například o univerzitní internetovou studovnu.

Mezi hlavní výhody patří nižší cena a spotřeba energie. V neposlední řadě je potřeba zmínit levnější servis, kdy při poruše kterékoliv stanice správce stanici pouze vymění bez potřeby instalace a konfigurace software.

Mezi nevýhody patří nutnost provozu na výkonné a spolehlivé síti. Pro některé aplikace může být obtížná realizace provozování multimédií na terminálech. Rovněž je nutné, aby terminálový a aplikační server byl velmi výkonný a zároveň spolehlivý a je nutné vyřešit jeho zálohování. Při výpadku serveru by všechny stanice byly nefunkční.

1.9.3 Instalace operačního systému ze serveru

Další možností je využít zavedení systému ze sítě pro nastartování instalačního programu, prostřednictvím kterého lze na stanici obsahující pevný disk nainstalovat běžný operační systém. Tento přístup umožňuje instalovat větší množství stanic bez využití instalace z CD nebo DVD, tím se instalace může zrychlit a automatizovat. Nevýhodou je potřeba serveru pro instalaci.

II. PRAKTICKÁ ČÁST

2 NASAZENÍ TERMINÁLOVÉ SÍTĚ

Nasazení terminálů je vhodné v případě nákupu nového počítačového vybavení nebo modernizace stávajícího vybavení. V obou případech je nutné zakoupit nový terminálový server dimenzovaný pro patřičný počet uživatelů.

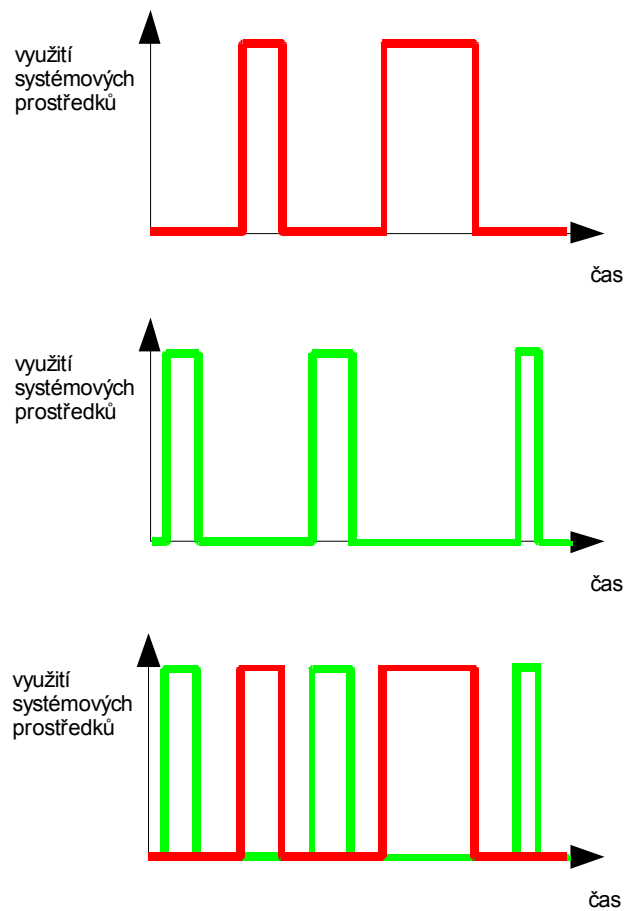
V případě využití stávajícího počítačového vybavení je možné využít stávající stanice jako terminály u uživatelů. Neměly by být využity stanice horší, než vybavené procesory třídy Pentium, vybavené méně než 32 MiB RAM, je rovněž možné z nich vyjmout pevné disky, optické a disketové mechaniky. Nutné je využití grafického a síťového adaptéru podporovaného v OS Linux, u síťového adaptéru je výhodou podpora standardu PXE. Pokud síťový adaptér nepodporuje PXE, je možné k zavedení systému využít stávající pevný disk, CompactFlash kartu, USB flashdisk případně zavedení z diskety nebo CD.

Výhodou tohoto řešení je ušetření většiny finančních prostředků za nákup nových stanic.

Nevýhodou je, že zejména starší počítače už překračují svou životnost a není možné garantovat přijatelnou spolehlivost.

V případě nákupu nového vybavení je nejvhodnější jako stanice použít speciální počítače (například: Fujitsu Siemens Futro nebo HP ThinClient), které jsou obvykle pro toto použití optimalizovány, a obsahují podporu všech potřebných standardů a podporu v OS Linux. Jejich nespornou výhodou je velmi nízká spotřeba (kolem 20W), malé rozměry a díky absenci pohyblivých dílů nízká hlučnost a poruchovost.

Nasazení sítě terminálů z pohledu ceny je v dlouhodobém horizontu (více jak 2 roky) výhodnější než nasazení plnohodnotných stanic. Úspory přináší zejména, že veškerý upgrade hardware a software se děje pouze na terminálovém serveru. Koncentrace velkého výpočetního výkonu do jednoho uzlu umožňuje jeho sdílení (agregaci) mezi více uživatelů, tím je možné ušetřit nemalé prostředky za elektrickou energii. Princip agregace mezi uživateli vysvětluje následující graf, viz: Obr. 3. Předpokládejme, že všechny počítače a server jsou stejně výkonné. Jedná se o závislost využití systémových prostředků v závislosti na čase. První graf znázorňuje využití systémových prostředků jedním uživatelem na své vlastní pracovní stanici. Druhý graf znázorňuje využití prostředků jiného uživatele na další stanici. Třetí graf znázorňuje využití systémových prostředků v případě, že by oba uživatelé využívali terminálový server jehož systémové prostředky by sdíleli.



Obr. 3: Využití systémových prostředků ve srovnání terminálového serveru a samostatných stanic

Nabízené řešení je vhodné taky z pohledu jednodušší a levnější správy software. Na stanicích stačí provozovat pouze minimalizovanou variantu Linuxu s X-serverem. Veškerá instalace nového software, aktualizací, upgrade probíhá pouze jednou na terminálovém serveru. Rovněž testování nasazení nových aplikací stačí provádět jen v souvislosti s konfigurací hardware a software, který obsahuje terminálový server.

Z pohledu bezpečnostních rizik je výhodou, že veškeré chyby v software je nutné záplatovat pouze na terminálovém serveru, rovněž je výhodou, že správa uživatelských účtů probíhá pouze na serveru.

Na UTB by terminálová síť mohla najít uplatnění například pro zpřístupnění Linuxu na pracovních stanicích pro studenty, nebo například pro výuku předmětů zaměřených na operační systémy, databáze a www technologie nebo pro vytvoření různých informačních

terminálů a kiosků. Ve všech případech by veškerá správa software, instalace a update mohly probíhat centrálně na serveru.

Výstavba terminálové sítě není dobrým rozhodnutím v okamžiku, kdy každý uživatel potřebuje mít neustále k dispozici výkonný počítač nebo v okamžiku, kdy uživatel potřebuje využívat hardware připojený přímo k jeho stanici (například vývojový kit mikropočítače).

Rovněž problematické by bylo využití při práci s multimédií (zejména přehrávání videa).

Naprostě nevhodná, z důvodu obtížné technické realizace, je v implementaci v oblastech využití hardwarové 3D akcelerace (například v oblasti počítačových her nebo počítačové grafiky).

2.1 Výběr používaného hardware

Jako terminálový server je vhodnější použít systém s vícejádrovým procesorem, nejlépe však víceprocesorový systém. Pro kancelářské použití by mělo stačit jeden procesor (respektive jedno jádro) na cca 5 až 10 uživatelů. I pro běžný kancelářský provoz serveru je potřeba ho vybavit dostatkem paměti, nejlépe je instalovat minimálně 512 až 1024 MiB pro základní systém a poté dalších 64 až 128 MiB (záleží na používaných aplikacích a na míře sdílení knihoven mezi aplikacemi) pro každého uživatele. Velikost instalovaného pevného disku by měla být minimálně 80 GB z toho 20 GB pro operační systém, programy a swap a zbylé místo pro data uživatelů, samozřejmě by měl být RAID. Pro více jak 15 až 20 uživatelů je vhodný 1Gb/s síťový adaptér pro připojení k síti terminálů a další 100Mb/s adaptéry pro připojení k internetu, intranetu případně k diskovému nebo poštovnímu serveru.

2.2 Výběr používaného software

Pro provoz terminálové sítě byl zvolen OS Linux z několika důvodů, jedním z hlavních je, že je k dispozici pod GNU/GPL licencí, která umožňuje libovolný rozsah úprav a prakticky neomezuje v rozsahu a způsobu použití systému.

Na terminál serveru byla zvolena distribuce CentOS 4.4, jedná se o klon RedHat Enterprise Linuxu sestavený z komponent šířených převážně pod licencí GNU/GPL, zaručuje to velkou míru kompatibility s komerční enterprise distribucí a vede k jednodušší

instalaci a provozu software jako Mathematica, Matlab nebo VmWare. Důležitá byla také široká škála dalších podporovaných aplikací z mnoha rozličných oborů a velká dostupnost technické podpory. Na terminálech bylo důležité, aby systém mohl být patřičně odlehčen a byl co nejvíce slučitelný s OS na terminálovém serveru, zvolen byl systém Slackware Linux 11. Jedná se o hardwarově nenáročnou, poměrně minimalistickou, distribuci.

Pro provoz boot serveru byl zvolen také Slackware, z důvodu minimální náročnosti na hardware a jeho jednoduchosti konfigurace.

2.3 Praktický start Linuxu ze sítě s využitím PXE

Při zapnutí běžného počítače třídy PC je prvním prováděným programem POST (Power On Self Test) který je součástí BIOS. Součástí BIOS bývá taky PXE. Pro použití PXE ke startu systému je nutné mít správně nastavené pořadí zařízení pro zavádění (boot sequence) a aktivovanu volbu zavádění ze sítě (boot from LAN).

PXE kód provede standardní dotaz na DHCP server, ten podle svého nastavení vrátí odpověď obsahující adresu klienta, síťovou masku, adresu brány, DNS serveru a adresu TFTP serveru. Odpověď DHCP serveru obsahuje také adresu souboru, který má PXE z TFTP serveru stáhnout.

PXE poté stáhne zadaný soubor a pokusí se jej spustit. V případě Linuxu může jít o zavaděč pxelinux, který je součástí balíku syslinux.

Pxelinux je takzvaný síťový zaváděcí program (network bootstrap program), který se stará o zavádění jednotlivých systémů. V případě pxelinux dojde ke stažení jádra Linuxu a obrazu inicializačního ramdisku prostřednictvím TFTP. Pro toto bývá využito parametrů získaných z DHCP serveru, které získal PXE. Pxelinux a další NBP obvykle také podporují zavedení systému mimo jiné také z místního pevného disku.

Celé spuštění OS Linux na terminálu je vhodné rozdělit do dvou fází, v první fázi proběhne stažení a zavedení jádra systému, a spuštění minimalizované verze systému z vytvořeného ramdisku. Minimalizovaná verze systému má za úkol nakonfigurovat síťový subsystém Linuxu a připojit síťový svazek z NFS serveru. Po připojení proběhne nahrazení kořenového systému souborů z ramdisku kořenovým systémem souborů ze síťového svazku. Po záměně následuje spuštění procesu init z nového kořenového systému souborů, tím program init přebere kontrolu nad systémem.

System je v tento okamžik v druhé fázi, ve které probíhá spuštění systémových služeb, konfigurace síťových rozhraní a k provádění startovacích skriptů umístěných v `/etc/rc.d`. Pokud je systém nakonfigurován tak, že výchozí runlevel je 4, pak start grafického rozhraní řídí skript `/etc/rc.d/rc.4`, skript musí být tedy upraven pro využití terminálového serveru. Příslušné úpravy jsou popsány dále.

2.4 Použitý hardware

Při testovacím provozu byla jako terminálový server využita sestava: AMD Athlon XP 1700+, 512 MiB RAM, 80GB pevný disk, 100Mb/s síťový adaptér a s integrovaným grafickým adaptérem nVidia GeForce4 MX.

Jako terminály a boot server byly použity sestavy: Intel Celeron 400MHz, 128MiB RAM, 10Mb/s síťový adaptér, grafický adaptér ATI Rage XL, pro provoz terminálu byly odpojeny pevné disky a využita CD-ROM jednotka pro zavedení PXE (projekt etherboot) z CD.

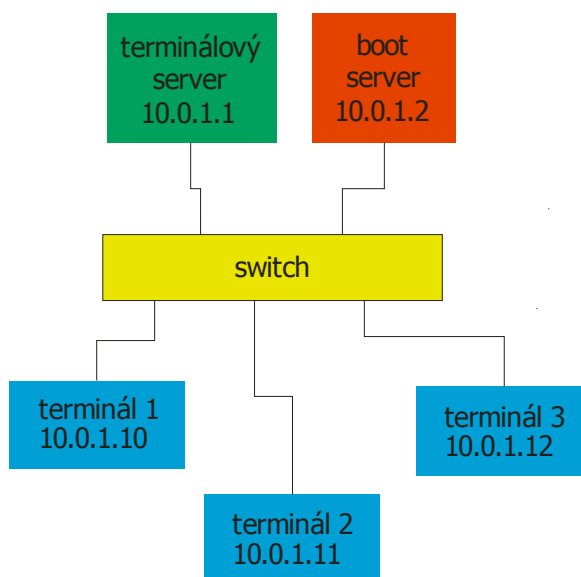
2.5 Použité síťové prvky

Pro propojení terminálů, boot serveru a terminálového serveru při testovacím provozu byl využit levný 5 portový switch s rychlostí 100 Mb/s. Menší levný switch postačoval z důvodu nízkého zatížení malým počtem stanic. Pro větší počet stanic by bylo vhodné použít kvalitní switch určený do 19" RACK skříně.

3 NÁVOD NA INSTALACI SERVERŮ A SESTAVENÍ SÍTĚ TERMINÁLŮ

3.1 Propojení terminálového serveru, boot serveru a terminálu do sítě

Servery a terminály propojte tzv. přímými kabely do použitého switchu. Topologie sítě bude tedy hvězdicová, (viz: Obr. 4).



Obr. 4: Zapojení jednoduché terminálové sítě

Před instalací zvolte pro terminálovou síť rozsah IP adres. V návodu je využit rozsah 10.0.1.0/24, čemuž odpovídají využitelné adresy 10.0.1.1 až 10.0.1.254 a síťová maska 255.255.255.0. Potřebné parametry (výchozí brány, DNS) pro zprovoznění linky do další sítě (například Internet) je důležité nastavit jen na terminálovém serveru. Všechny parametry pokud možno konzultujte se svým správcem sítě.

3.2 Instalace a nastavení terminálového serveru

Pro jednoduchost budou popsány odlišnosti od výchozí instalace. Instalace operačního systému CentOS 4.4 na terminálový server proveďte z oficiálního instalačního DVD. Instalaci začněte vložením DVD do mechaniky a restartujte počítač (je nutné mít aktivovanou podporu zavádění z CD/DVD v BIOS). Po skončení POST následuje spuštění instalátoru.

Jako jazyk pro instalaci zvolte: **český**. Jako typ instalace zvolte: **pracovní stanice**. Při rozdělování disku rozdělte disk pomocí nástroje **Disk Druid** na 1GB oddíl pro swap, 20GB oddíl pro systém (bod připojení /) a zbylý prostor přidejte pro domovský adresář (bod připojení /home). Při nastavení sítě zvolte název a doménu serveru a u nastavení síťové karty vypněte podporu DHCP. IP adresu a masku sítě nastavte na 10.0.1.1 a 255.255.255.0, výchozí bránu a adresy DNS nastavte podle pokynů správce sítě nebo poskytovatele připojení. U položky **firewall** povolte (zatrhněte): **vzdálené přihlášení (SSH)**. Ve volbě instalovaných balíčků zvolte: **upravit sadu instalovaných balíčků software**. Ve volbě výběru skupin balíčků zatrhněte navíc volby: **KDE** a **nástroje pro konfiguraci serveru**.

Po instalaci a restartu počítače následuje konfigurace nainstalovaného systému. V nastavení data a času v záložce **NTP** povolte NTP a jako server přidejte například server **tik.cesnet.cz**. Nastavení zobrazení nemá žádný vliv na nastavení zobrazení na terminálu, nastavení tedy přizpůsobte potřebám na serveru. Jako dalšího uživatele v systému si pro správu vytvořte další účet, protože přihlášení jako root je jistým bezpečnostním rizikem, rovněž máte možnost nastavení síťového přihlašování.

Po konfiguraci je systém připraven k použití jako lokální stanice.

Pro použití jako terminálový server je nutné povolit síťový provoz XDM. V CentOS 4.4 je jako XDMCP server použit GDM.

V souboru `/etc/X11/gdm/gdm.conf` povolte vzdálené přihlášení. V sekci `[xdmcp]`

Nastavte: `Enabled=true`

Další nastavení se vztahují k tomtéž souboru. Výpis změn je proveden podle sekcí (sekce jsou uzavřeny v hranatých závorkách).

GNOME má dva typy tzv. uvítacích obrazovek. Jeden typ se je využít pro lokální přihlašování (více graficky náročný) a druhý pro vzdálené přihlašování (jednodušší verze).

V sekci `[daemon]` nastavte:

```
Greeter=/usr/bin/gdmgreeter
```

```
RemoteGreeter=/usr/bin/gdmlogin
```

Z důvodů bezpečnosti lze ovlivnit povolení přihlašování uživatele root a to jak pro lokální přihlášení, tak pro vzdálené. V sekci [security] nastavte:

```
AllowRoot=true
```

```
AllowRemoteRoot=false
```

Na zatížení serveru má vliv maximální počet otevřených sezení (terminálů v provozu).

V sekci [xdmcp] nastavte:

```
MaxSession=16
```

U terminálového serveru je žádoucí zakázat možnost vzdáleného vypnutí, restartu nebo suspendování počítače. Je taky možné změnit uvítací pozdrav u přihlašovací obrazovky.

V sekci [greeter] nastavte:

```
SystemMenu=false
```

```
Welcome=Welcome to %n
```

```
RemoteWelcome=Welcome to %n
```

Tímto je nastavení GDM dokončeno.

Zbývá povolit na firewallu službu XDMCP. Pro editaci nastavení firewallu se přihlaste do systému, zvolte nabídku v levém dolním rohu obrazovky. Dále vyberte **systemová nastavení** a položku **úroveň zabezpečení**. V dialogovém okně do řádku ostatní porty zadejte řetězec: **177:udp**.

Pro aplikaci všech nastavení server restartujte. Po restartu bude terminálový server připraven k použití.

3.3 Instalace a nastavení boot serveru

Pro jednoduchost budou popsány odlišnosti od výchozí instalace. Kompletní seznam instalovaných balíčků je v příloze P I. Instalace operačního systému Slackware Linux 11 na boot server provedte z oficiálních instalačních CD. Instalaci začnete vložením CD do mechaniky a restartujte počítač (je nutné mít aktivovanu podporu zavádění z CD/DVD v BIOS). Po skončení POST následuje spuštění zavaděče z instalačního CD.

Po spuštění instalátoru je zobrazena žádost o přihlášení jakoby se jednalo o běžný systém. Přihlaste se jako uživatel root s prázdným heslem.

Při instalaci vyberte jako rozložení klávesnice (**keyboard layout**) například volbu **qwerty-cz-lat2.map**. Rozdělení disku proveďte pomocí programu fdisk na například 256MB pro swap a minimálně 2GB pro kořenový oddíl. Program na rozdělení disku spusťte příkazem `fdisk /dev/hda`, kde hda znamená nastavení pevného disku master na primárním rozhraní IDE, hdb znamená slave na primárním rozhraní IDE atd. Program při editaci tabulky oddílů využívá klávesové zkratky. Pro zobrazení nápovědy stiskněte klávesu **m**. Zobrazení aktuálního rozdělení disku probíhá pomocí klávesy **p**. Případné existující oddíly odstraní pomocí klávesy **d**. Nový oddíl vytvoříte stisknutím **n**, v následujícím dialogu program vyzve ke zvolení čísla oddílu a k zadání prvního a posledního cylindru, případně k zadání prvního cylindru a velikosti oddílu. Oddíly číslyte v pořadí 1, 2, 3 atd. Výchozí typ vytvořeného oddílu je typ Linux, pro změnu na Linux swap zvolte **t** a zadejte číslo 83. Ukončení rozdělení disku a aplikování změn proveďte volbou **w**.

Pro pokračování instalace spusťte instalační program příkazem `setup`.

V instalaci pokračujte aktivací swap oddílu a určením cílového diskového oddílu.

Následuje výběr balíčků a instalace systému na pevný disk. Při výběru balíčků vyberte skupiny aplikací: **A**, **AP**, **L**, **N**. Jako režim výběru balíčků zvolte pomocí **expert**. Následující balíčky vyberte podle seznamu balíčků v příloze P I.

Po instalaci balíčků následuje finální konfigurace systému. Ve finální konfiguraci volte výchozí hodnoty, vyjma zavaděče LILO a nastavení sítě. U nastavení LILO zvolte jako cíl volbu **MBR**, protože LILO bude hlavním zavaděčem. U nastavení sítě upravte výchozí nastavení podle zvoleného rozsahu adres pro terminálovou síť tak, aby adresa serveru byla nastavena napevno (nebyla přidělována přes DHCP). Nastavte tedy například IP adresu 10.0.1.2 a síťovou masku 255.255.255.0. Síťovou bránu a DNS nastavte stejně jako u terminálového serveru.

Po nastavení hesla uživatele root podle instrukcí instalátoru vyjměte instalační médium a restartujte server.

Po restartu nově nainstalovaného serveru se přihlaste jako uživatel root.

Protože na boot serveru má být spuštěna služba TFTP nainstalujte balíček obsahující démona xinetd. Pro Slackware Linux bývá dostupný balíček například na www.linuxpackages.net. Proveďte jeho stažení pomocí příkazu:

```
wget ftp://ftp.slackware.hu/linuxpackages/Slackware-11.0/Daemon/xinetd/xinetd-2.3.14-i486-1bms.tgz
```

Pro zprovoznění služby TFTP vytvořte konfigurační soubor `/etc/xinetd.conf`, který bude obsahovat následující:

```
defaults
{
instances = 60
log_type = SYSLOG authpriv
log_on_success = HOST PID
log_on_failure = HOST
cps = 25 30
}
includedir /etc/xinetd.d
```

Poslední řádek konfiguračního souboru znamená, že další konfigurační soubory pro jednotlivé služby jsou umístěny v adresáři `/etc/xinetd.d`. V adresáři `/etc/xinetd.d` vytvořte soubor s názvem `tftp`. Ten bude obsahovat:

```
service tftp
{
socket_type = dgram
protocol = udp
wait = yes
user = root
server = /usr/sbin/in.tftpd
server_args = -s /tftpboot
per_source = 11
cps = 100 2
flags = IPv4
}
```

Důležité volby jsou: `server` a `server_args`. Volba `server` odkazuje na spustitelný soubor TFTP serveru a volba `server_args` na pracovní adresář serveru (adresář s poskytovanými daty). Při zadávání veškerých cest týkajících se TFTP (konfigurační soubory PXELinux zavaděče) zadávejte adresy bez části týkající se cesty k pracovnímu adresáři TFTP.

Dále nakonfigurujte DHCP server. Vytvořte soubor `/etc/dhcpd.conf`, který bude obsahovat:

```
ddns-update-style ad-hoc;
allow booting; #podpora startu ze site
allow bootp;

subnet 10.0.1.0 netmask 255.255.255.0 { #adresa a rozsah site
option domain-name "terminal-net";      #prirazena domena
option subnet-mask 255.255.255.0;      #sitova maska
option broadcast-address 10.0.1.255;    #vsesmerova adresa
option domain-name-servers 10.0.1.2;    #DNS server
option routers 10.0.1.2;                #vychozi brana
}

# Seskupeni PXE klientu
group {
# konfigurační direktivy pro PXE...
next-server 10.0.1.2; #adresa TFTP serveru
filename "/pxelinux.0"; #soubor stahovany z TFTP
# Pokud nepouzivame dynamicke adresace,
# budeme potrebovat zaznam jako tento pro kazdy terminal
    host stanice1
    {
        hardware ethernet 00:0C:29:D6:E3:4F; #MAC terminalu
        fixed-address 10.0.1.10;            #prirazena IP
    }

#konec zaznamu terminalu

    host stanice2
    {
        hardware ethernet 00:0C:29:D6:E3:5D; #MAC terminalu
        fixed-address 10.0.1.11;            #prirazena IP
    }

#
}
```

Při konfiguraci DHCP serveru je nutné zohlednit použitý rozsah IP adres v síti. Pro výstavbu terminálové sítě využijte adresy z adresního rozsahu třídy A, kde jsou k dispozici adresy 10.0.0.0 až 10.255.255.255, nebo rozsahu třídy B, kde jsou k dispozici adresy 172.16.0.0 až 172.31.255.255, nebo rozsah třídy C, kde jsou k dispozici adresy 192.168.0.0 až 192.168.255.255.

Řádky konfiguračního souboru za znakem „#“ jsou komentáře. Většina voleb je popsána komentáři přímo v konfiguračním souboru. Pro případ využití jiného adresního rozsahu upravte volby `subnet`, `netmask`, `routers`, `subnet-netmask`, `domain-name-servers`, `next-server` a pak upravte volby pro záznamy terminálu.

Každý terminál by měl obsahovat v konfiguračním souboru DHCP vlastní záznam.

Pokračujte zprovozněním služby NFS editací souboru `/etc/exports`. Soubor bude obsahovat pouze řádek: `/diskless 10.0.1.*(rw,no_root_squash)`

Řetězec `10.0.1.*` znamená že na poskytovaný svazek budou mít přístup stanice s IP adresami 10.0.1.0 až 10.0.1.255, tento řetězec upravte podle adresního prostoru dostupného pro terminálovou síť. Řetězec `(rw,no_root_squash)` určuje, jako povolené akce čtení a zápis a nepovoluje přenášení práv uživatele `root` pro svazek, tento řetězec neupravujte.

Pokračujte vytvořením adresáře `/diskless`, do vytvořeného adresáře zkopírujete obsah kořenového oddílu systému, jehož vytvoření je popsáno níže.

Spuštění dhcp serveru probíhá příkazem `/usr/sbin/dhcpd` a spuštění xinetd pomocí `/usr/sbin/xinetd`. Proto tyto příkazy přidejte do skriptu `/etc/rc.d/rc.local` (každý příkaz na samostatný řádek), přidání zajistí spuštění `dhcpd` a `xinetd` při startu.

3.4 Vytvoření systému provozovaného na bezdiskových terminálech

Postup se skládá z vytvoření části, která bude přes síť stažena a spuštěna v terminálu a z části, která bude na boot serveru a terminál ji připojí jako svůj kořenový oddíl.

Jako základ použijte Slackware pro jednoduchost jeho balíčkovacího systému, konfiguračních souborů a startovacích skriptů. Pro vytvoření systému pro terminál nainstalujte systém nejprve běžným způsobem na pevném diskem vybaveného tenkého klienta nebo do virtuální stanice například ve VMware Workstation. Rovněž by bylo

možné vystavět systém přímo na boot serveru z jeho systému. Naproti tomu vytváření mimo boot server dává možnost testování v průběhu prací (například různých konfigurací jádra).

V případě využití dalšího počítače je průběh instalace stejný jako u instalace boot serveru, rozdíl je pouze ve skupinách instalovaných balíčků a v instalovaných balíčcích (seznam popsán v příloze P II). Ze skupin balíčků zvolte: **A,AP,D,L,N,X**.

V případě využití boot serveru vyberte při jeho instalaci navíc skupinu balíčků **D** a nainstalujte balíčky popsané v příloze vztahující se k boot serveru a zároveň balíčky vztahující se k počítači pro vytvoření distribuce pro bezdiskový terminál.

Po úspěšné instalaci a restartu následuje spuštění nainstalovaného systému.

3.4.1 Vytvoření části systému určené pro provoz na terminálu

Jako uživatel root se přihlaste systému.

Nyní z internetu stáhněte dvě součásti (resp. jejich zdrojové kódy), které budete potřebovat pro vytvoření systému provozovaného na terminálech. Jsou jimi: Linux Kernel (domovská stránka: www.kernel.org), BusyBox (domovská stránka: www.busybox.net) a udhcp (domovská stránka udhcp.busybox.net).

Použít můžete následující postup:

Přesunutí do pracovního adresáře:

```
cd /usr/src
```

Stažení jádra Linux:

```
wget http://kernel.org/pub/linux/kernel/v2.6/linux-  
2.6.20.1.tar.bz2
```

Stažení BusyBox:

```
wget http://busybox.net/downloads/busybox-1.4.1.tar.bz2
```

Stažení udhcp:

```
wget http://udhcp.busybox.net/source/udhcp-0.9.8.tar.gz
```

Práce začněte sestavením a přeložením jádra operačního systému Linux, které poběží na terminálu.

Rozbalení jádra:

```
tar -jxvf linux-2.6.20.1.tar.bz2
```

Přesunutí do adresáře s jádrem:

```
cd linux-2.6.20.1
```

Spuštění konfiguračního nástroje:

```
make menuconfig
```

V zobrazeném menu máte možnost zvolit, které ovladače budou přeloženy zároveň s jádrem systému, rovněž máte možnost volit parametry ovladačů a parametry systému (architektura a typ procesoru, podpora více procesorů, výkonnostní parametry).

Při výběru ovladačů je potřeba aby všechny ovladače byly do jádra systému vloženy přímo (ne jako externí moduly), toho docílíme pomocí volby * u ovladače. Je to vhodné z důvodu nedostupnosti pevného disku během provozu terminálu, jiným řešením by bylo umístění modulů (volba **M**) do obrazu ramdisku spolu se základním systémem souborů. Řešení s ovladači v ramdisku je vhodné zejména v okamžiku kdy by v síti bylo využito více různých hardwarových konfigurací terminálů a jádro obsahující ovladače pro veškerá zařízení by bylo příliš velké (v paměti by obsadilo zbytečně moc prostoru). Pro funkci terminálu jsou klíčové ovladače pro ramdisk, systémy souborů NFS, Ext2/3 a ReiserFS, použité síťové adaptéry a pro uživatelská rozhraní (klávesnice a myš). Vhodné je zařadit podporu IPtables a IDE rozhraní pevných disků.

Z důvodu rozsáhlosti konfigurace je kompletní konfigurační soubor v příloze P III. Jedná se o soubor `.config`.

Spuštění překladu jádra:

```
make
```

Po úspěšném překladu najdete výsledný binární soubor jádra a ovladačů v adresáři `/usr/src/linux-2.6.20.1/arch/i386/` z toho jádro v podadresáři `boot`.

Následuje návrat do pracovního adresáře `/usr/src` a integrace udhcp do BusyBox:

```
cd /usr/src
```

Rozbalení BusyBox:

```
tar -jxvf busybox-1.4.1.tar.bz2
```


Rozbalení udhcp:

```
tar -jxvf udhcp-0.9.8.tar.gz
```

Přesunutí do adresáře s udhcp:

```
cd udhcp-0.9.8
```

Zkopírování udhcp do BusyBox:

```
cp *.[ch] ../busybox-1.4.1/networking/udhcp
```

Přesunutí do adresáře s BusyBox:

```
cd ../busybox-1.4.1
```

Spuštění konfiguračního nástroje:

```
make menuconfig
```

V konfiguračním nástroji zapněte následující volby podle seznamu popsaného v příloze P IV. Případně využijte soubor `.config` z přílohy.

spuštění překladu busybox:

```
make
```

Instalace do adresáře pro vytvoření ramdisku:

```
make PREFIX=/boot/initrd-tree install
```

V adresáři `/boot/initrd-tree` naleznete adresářovou strukturu budoucího systému (z části vytvořena pomocí instalace BusyBox).

Následuje vytvoření souboru `linuxrc`. Může jít o binární soubor nebo shellovský skript. Kompletní skript použitý pro terminál naleznete v příloze P V. Vytvořte tento skript pomocí vašeho oblíbeného textového editoru v adresáři `/boot/initrd-tree`.

Vytvoření komprimovaného obrazu ramdisku:

```
mkinitrd
```

Výsledný komprimovaný obraz naleznete pod názvem `initrd.gz` v adresáři `/boot`.

Obraz ramdisku je v systému souborů Ext2 a je komprimován pomocí gzip. Pro editaci a experimentování s obrazem je možné ho připojit a pracovat s ním jako s běžným pevným diskem.

Nejprve před připojením obraz rozbalte:

```
tar -jxvf initrd.gz
```

Obraz připojte pomocí příkazu:

```
mount -o loop -t ex2 initrd /mnt/ramdisk
```

Odpojení proved'te příkazem:

```
umount initrd
```

Nyní máte k dispozici jádro systému – vmlinuz a komprimovaný obraz ramdisku – initrd.gz obsahující kořenový systém souborů, základní utility, skripty a volitelně ovladače.

Tyto soubory nakopírujte do adresáře /tftpboot/slackware na boot serveru.

Příkazy pro zkopírování jádra a obrazu ramdisku v případě že veškeré přípravy byly prováděny na boot serveru:

```
cp /usr/src/linux-2.6.20.1/arch/i386/boot/vmlinuz  
/tftpboot/slackware
```

```
cp /boot/initrd.gz /tftpboot/slackware
```

Do adresáře /tftpboot nakopírujte zavaděč pxelinux pomocí příkazu:

```
cp /usr/share/syslinux/pxelinux.0 /tftpboot
```

Zavaděč pxelinux hledá konfigurační soubory v adresáři /tftpboot/pxelinux.cfg, proto tento adresář vytvořte a ve vytvořeném adresáři vytvořte konfigurační soubor pxelinuxu pro stanice. Soubor bývá pojmenován buď podle MAC adresy stanice kde název souboru bude začínat 01- a dále adresou zapsanou v hexadecimálním formátu malými písmeny kde jsou dvojice znaků odděleny pomlčkami, nebo podle IP adresy terminálu, a to tak, že je adresa zapsána v hexadecimálním formátu bez teček a velkými písmeny. V případě MAC adresy 00:11:22:33:44:55 bude název souboru 01-00-11-22-33-44-55. Například pokud bude IP adresa terminálu 10.0.1.10 tak název konfiguračního souboru vztahujícího se k terminálu bude: 0A00010A. Pro větší rozsahy IP adres jako například 10.0.1.0 až 10.0.1.255 je možné soubor pojmenovat 0A0001, v názvu je vynechána měnící se část adresy. Pokud není nalezen patřičný konfigurační soubor pak pxelinux hledá soubor s názvem default. V případě homogenní konfigurace terminálů stačí vytvořit soubor s názvem default.

Vytvořený soubor bude vypadat takto:

```
default std
```

```
label std
```

```
kernel slackware/vmlinuz
```

```
append initrd=slackware/initrd.gz ROOTDEV=10.0.1.2:/diskless/  
root=/dev/ram0
```

Parametr `kernel` udává cestu k jádru, parametr `append` udává parametry jádra, například cestu ke zabalenému obrazu ramdisku, cestu k počátečnímu kořenovému oddílu a obsah proměnné `ROOTDEV`. Přes proměnnou `ROOTDEV` je skriptu `linuxrc` předána adresa NFS svazku, který má být připojen jako budoucí kořenový oddíl. Předávání parametrů má výhodu v tom, že může existovat jeden ramdisk, který není nutné upravovat například při změně adresy NFS serveru a podobně.

3.4.2 Vytvoření části připojované z boot serveru

Na boot server zkopírujte do adresáře `/diskless` z počítače, kde jste vytvářeli systém pro provoz terminálu následující adresáře: `/bin /dev /etc /lib /opt /sbin /usr /var` a vytvořte adresáře `mnt, old_root, proc, root, sys` a `tmp`.

Zkopírování provedte například pomocí NFS následujícím způsobem.

Připojení vzdáleného svazku:

```
mount 10.0.1.2:/diskless/ /mnt/floppy
```

Nastavení aktuální pozice do kořenového adresáře:

```
cd /
```

Zkopírování adresářů:

```
cp -R bin /mnt/floppy
```

```
cp -R dev /mnt/floppy
```

```
cp -R etc /mnt/floppy
```

```
cp -R lib /mnt/floppy
```

```
cp -R opt /mnt/floppy
```

```
cp -R sbin /mnt/floppy
```

```
cp -R usr /mnt/floppy
```

```
cp -R var /mnt/floppy
```

Vytvoření zbylých adresářů

```
mkdir /mnt/floppy/mnt
```

```
mkdir /mnt/floppy/old_root
```

```
mkdir /mnt/floppy/proc
```

```
mkdir /mnt/floppy/root
```

```
mkdir /mnt/floppy/sys
```

```
mkdir /mnt/floppy/tmp
```

Odpojení vzdáleného svazku:

```
umount 10.0.1.2:/diskless/
```

Nyní by měl terminál být schopen zavést systém ze sítě, připojit vzdálený svazek a zobrazit žádost o přihlášení zatím v textovém režimu. Přihlaste se jako uživatel root.

Po spuštění máte k dispozici na terminálu kompletně vybavený systém včetně vývojových nástrojů a podobně. Nepotřebné součásti můžete odstranit, jejich přítomnost však ničemu nevadí. Může se jednat o obsah adresáře `/diskless/usr/src` a o vývojové nástroje (skupina balíčků D). Obsah adresáře `/usr/src` odstraňte smazáním.

```
cd /diskless/usr/src
```

```
rm -Rf *
```

Pro odebrání nepotřebných balíčků po spuštění terminálu použijte nástroj `pkgtool` a v menu zvolte **Remove**. V zobrazeném dialogovém okně zvolte k odebrání balíčky: **gcc**, **gcc-g++**, **gcc-objc**, **gd**, **gdbm**, **make** a potvrďte odebrání.

Nyní můžete ověřit správnou funkci grafického prostředí v návaznosti na terminálový server pomocí příkazu: `xinit -query 10.0.1.1` kde 10.0.1.1 je adresa terminálového serveru.

Pro spuštění grafického rozhraní a připojení na server při startu systému proveďte následující úpravy týkající se výchozího runlevelu terminálu a startovacího skriptu na terminálu, tyto soubory jsou ve skutečnosti uloženy na boot serveru v adresáři `/diskless`. Změna výchozího runlevelu spočívá v editaci souboru

/diskless/etc/inittab, na řádku obsahujícím: id:3:initdefault: změňte z 3 na 4. Změna na výchozí runlevel 4 má za následek mimo jiné provedení skriptu /diskless/etc/rc.d/rc.4. Stávající obsah skriptu rc.4 nahraďte proto následujícím obsahem, kde 10.0.1.1 je adresa terminálového serveru.

```
if [ -x /usr/X11R6/bin/X ]; then
exec /usr/X11R6/bin/X -query 10.0.1.1
fi
```

Po této úpravě bude terminál schopen zavést systém ze sítě, spustit X server a připojit se na terminálový server.

Terminálová síť a servery je kompletní a plně funkční.

ZÁVĚR

V práci je popsán způsob a možnosti zavedení operačního systému Linux ze síťového prostředí. Start systému ze sítě může najít využití jak k vytvoření sítě nejrůznějších terminálů nebo bezdiskových stanic, tak i například v průmyslových a embedded (vestavěné počítače, například routery, atd.) zařízeních. V práci je pro start systému Linux a provoz terminálové sítě využita síť typu Ethernet.

Při práci na problematice vznikla řada drobných problémů, které se povedlo postupně odstranit. Nejzávažnější problém bylo vytvořit systém na terminálu tak, aby byl co nejmenší a jeho provoz zatěžoval síť co nejméně, a zároveň aby obsahoval X server schopný spolupráce s terminálovým serverem. Tento problém se povedlo vyřešit tak, že systém je zaveden ve dvou fázích. V první fázi je zavedeno jádro a staženo několik málo nástrojů pro zprovoznění síťového připojení. Dále proběhne připojení nového svazku z NFS a záměna stávajícího kořenového oddílu za nově připojený svazek. V druhé fázi systém pokračuje spuštěním programu *init*, který převezme kontrolu nad systémem a zajistí provedení startovacích skriptů, jakoby se jednalo o systém instalovaný na pevném disku. Součástí skriptů pro start systému je mimo jiné taky start X serveru a jeho připojení na terminálový server.

ZÁVĚR V ANGLIČTINĚ

In this work are described ways and possibilities of the Linux operating system started from network. It can be applied in a network of terminals and stations without hard drives. It can be also used in industrial and embedded devices. In this work, there is used Ethernet for starting the Linux system and running the network of terminals.

Many little problems appeared during the analysis. Finally all of them were progressively solved. The main task was to make a system at a terminal, which should be the smallest and not very demanding. It should also contain the X server, which is able to cooperate with a terminal server. To fix the problem, the system boots in two phases. The kernel is loaded at first. Also other tools for network connection are loaded. Then a new network drive from NFS is mapped and the current root part changed for the new drive. System starts up a program init, which takes control over. It provides running starting scripts in the same way as a system installed at a hard drive. A part of the scripts is also a start of the X server and his connection to the terminal server.

SEZNAM POUŽITÉ LITERATURY

- [1] DEITEL, HARVEY, M. *Operating Systems*. [s.l.] : Prentice Hall, 2004. 1100 s.
- [2] TANENBAUM, Andrew S. *Modern operating systems*. [s.l.] : Prentice Hall, c2002. 951 s.
- [3] *Linux - Dokumentační projekt*. Brno : Computer Press, 2003. 1001 s.
- [4] SOBELL, Mark G. *Linux - Praktický průvodce*. Brno : Computer Press, 1999. 946 s.
- [5] NEMETH, Evi, SNYDER, Garth, HEIN, Trent R. *Linux - Kompletní příručka administrátora*. Brno : Computer Press, 2004. 880 s.
- [6] Linux [online]. 2007 [cit. 2007-04-25]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Linux>>.
- [7] Preboot Execution Environment [online]. 2007 [cit. 2007-02-17]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Preboot_Execution_Environment>.
- [8] X.org [online]. 2007 [cit. 2007-02-17]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/X.org>>.
- [9] X display manager [online]. 2007 [cit. 2007-02-17]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Xdmcp>>.
- [10] Dynamic Host Configuration Protocol [online]. 2007 [cit. 2007-02-20]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Dhcp>>.
- [11] Trivial File Transfer Protocol [online]. 2007 [cit. 2007-02-20]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Tftp>>.
- [12] Network File System [online]. 2006 [cit. 2007-04-25]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Network_File_System>.
- [13] VALOUŠEK, Ondřej. *Jak nabootovat Linux po síti* [online]. 2005 [cit. 2007-02-10]. Dostupný z WWW: <<http://www.abclinuxu.cz/clanky/navody/jak-nabootovat-linux-po-siti>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

- b, bit [:bit:] Jedná se o základní jednotku informace, rovněž jde o nejmenší možnou jednotku informace, nabývá dvou stavů: pravda nebo nepravda (true or false), značených taky 1 a 0 nebo High a Low.
Použití: S předponami k,M,G a časovým údajem udává propustnost datových sítí nebo seriových sběrnic. Mimo jiné udává také šířku paralelních sběrnic.
- B, byte [:bajt:] Jde o násobek základní jednotky informace (někdy označován jako slovo). 1B obsahuje 8b. Může nabývat až 2^8 (tedy 256) stavů.
Použití: mimo jiné s předponami k,M,G nebo ki,Mi,Gi udává kapacitu paměťového prostoru.
- M,G (MB,GB) Předpony k,M a G jsou předponami ze soustavy jednotek SI. Jejich základem je 10, platí tedy: $k=10^3$, $M=10^6$, $G=10^9$. V kombinaci s jednotkou informace se používají k označení většího objemu dat. Někdy bývají chybně zaměňovány předpony k,M,G s ki,Mi,Gi (například OS Windows takto chybně informují o objemech dat).
Příklad: 20GB = 20 000 000 000 B
- Mi,Gi (MiB,GiB) Jedná se o předpony, jejichž základem je 2^{10} ($2^{10}=1024$), nacházejí uplatnění zejména v oblasti výpočetní techniky. Platí tedy:
 $ki=2^{10}=1024$,
 $Mi=2^{20}=1\,048\,576$, $Gi=2^{30}=1\,073\,741\,824$
Příklad: 20GiB = 21 474 836 480 B
- Mb/s,Gb/s Jednotky přenosové rychlosti v počítačových sítích.

SEZNAM OBRÁZKŮ

| | |
|---|----|
| Obr. 1: Princip funkce X serveru v lokálním a síťovém prostředí | 13 |
| Obr. 2: Princip funkce terminálu a terminálového serveru..... | 14 |
| Obr. 3: Využití systémových prostředků ve srovnání terminálového serveru a samostatných stanic | 20 |
| Obr. 4: Zapojení jednoduché terminálové sítě..... | 24 |

SEZNAM PŘÍLOH

- P I: Seznam balíčků instalovaných na boot server.
- P II: Seznam balíčků instalovaných pro vytvoření terminálu.
- P III: Konfigurační soubor pro překlad jádra Linuxu.
- P IV: Konfigurační soubor pro překlad nástrojů BusyBox.
- P V: Skript pro start terminálu.
- P VI: CD obsahující elektronickou verzi bakalářské práce a konfigurační soubory.

PŘÍLOHA P I: SEZNAM BALÍČKŮ INSTALOVANÝCH NA BOOT SERVER

Bližší popis balíčků lze získat na adrese <http://packages.slackware.it/> po zadání názvu balíčku a verze Slackware.

| | | | |
|-------------|----------------|--------------|---------------|
| aaa_base | findutils | logrotate | reiserfsprogs |
| aaa_elflibs | gawk | lsof | rsync |
| acl | gettext | m4 | rzip |
| acpid | gettext-tools | man | screen |
| apmd | getty-ps | man-pages | sed |
| attr | glib | mc | shadow |
| autoconf | glib2 | mercurial | shared-mime- |
| automake | glibc | mhash | info |
| bash | glibc-i18n | minicom | slocate |
| bc | glibc-profile | mm | smartmontools |
| bin | glibc-solibs | module-init- | sudo |
| binutils | glibc-zoneinfo | tools | sysfsutils |
| bison | gpm | most | sysklogd |
| bpe | grep | nasm | syslinux |
| byacc | groff | nc | sysstat |
| bzip2 | gzip | ncftp | sysvinit |
| ccache | hdparm | ncurses | t1lib |
| coreutils | hotplug | nfs-utils | tar |
| cpio | indent | nmap | tcpdump |
| cscope | inetd | ntp | tcpip |
| cvs | iproute2 | openssh | texinfo |
| cxxlibs | iptables | pciutils | traceroute |
| dcrn | isapnptools | pine | udev |
| devs | kbd | pkgconfig | utempter |
| dhcpcd | kernel-ide | pkgtools | util-linux |
| diffstat | kernel-modules | popt | wget |
| diffutils | less | portmap | zlib |
| e2fsprogs | libidn | procps | zsh |
| elvis | libtool | quota | |
| etc | lilo | readline | |

PŘÍLOHA P II: SEZNAM BALÍČKŮ INSTALOVANÝCH PRO VYTVOŘENÍ TERMINÁLU

Bližší popis balíčků lze získat na adrese <http://packages.slackware.it/> po zadání názvu balíčku a verze Slackware.

| | | | |
|-------------|----------------|---------------|-----------------|
| aaa_base | gcc-g++ | man | shadow |
| aaa_elflibs | gcc-objc | man-pages | shared-mime- |
| acl | gd | mc | info |
| acpid | gdbm | mercurial | slocate |
| apmd | gettext | mhash | smartmontools |
| attr | gettext-tools | minicom | sudo |
| autoconf | getty-ps | mkinitrd | sysfsutils |
| automake | glib | mm | sysklogd |
| bash | glib2 | module-init- | syslinux |
| bc | glibc | tools | sysstat |
| bin | glibc-i18n | most | sysvinit |
| bin86 | glibc-profile | nasm | t1lib |
| binutils | glibc-solibs | nc | tar |
| bison | glibc-zoneinfo | ncftp | tcpdump |
| bpe | gpm | ncurses | tcpip |
| byacc | grep | nfs-utils | texinfo |
| bzip2 | groff | nmap | traceroute |
| ccache | gzip | ntp | udev |
| coreutils | hdparm | openssh | utempter |
| cpio | hotplug | pciutils | util-linux |
| cscope | indent | pcre | wget |
| cvs | inetd | perl | x11 |
| cxxlibs | iproute2 | pine | x11-devel |
| dcrn | iptables | pkgconfig | x11-docs |
| devs | isapnptools | pkgtools | x11-docs-html |
| dhcpcd | kbd | pmake | x11-fonts- |
| diffstat | kernel-headers | popt | 100dpi |
| diffutils | kernel-ide | portmap | x11-fonts- |
| e2fsprogs | kernel-modules | procps | cyrillic |
| elvis | less | python | x11-fonts-misc |
| etc | libidn | quota | x11-fonts-scale |
| findutils | libtool | readline | x11-xdmx |
| freetype | lilo | reiserfsprogs | x11-xnest |
| gawk | logrotate | rsync | x11-xvfb |
| gcc | lsof | rzip | zlib |
| | m4 | screen | zsh |
| | make | sed | |

PŘÍLOHA P III: KONFIGURAČNÍ SOUBOR PRO PŘEKLAD JÁDRA LINUXU

Pro jednoduchost nejsou zobrazeny zakomentované řádky. Jde o soubor `/usr/src/linux-2.6.20.1/.config`. Tento soubor lze dále editovat nejlépe pomocí příkazu `make menuconfig`, soubor pro editaci musí být umístěn v adresáři `/usr/src/linux-2.6.20.1` a i spuštění příkazu musí proběhnout v tomtéž adresáři.

```
CONFIG_X86_32=y
CONFIG_GENERIC_TIME=y
CONFIG_LOCKDEP_SUPPORT=y
CONFIG_STACKTRACE_SUPPORT=y
CONFIG_SEMAPHORE_SLEEPERS=y
CONFIG_X86=y
CONFIG_MMU=y
CONFIG_GENERIC_ISA_DMA=y
CONFIG_GENERIC_IOMAP=y
CONFIG_GENERIC_HWEIGHT=y
CONFIG_ARCH_MAY_HAVE_PC_FDC=y
CONFIG_DMI=y
CONFIG_DEFCONFIG_LIST="/lib/modules/$UNAME_RELEASE/.config"
CONFIG_EXPERIMENTAL=y
CONFIG_BROKEN_ON_SMP=y
CONFIG_LOCK_KERNEL=y
CONFIG_INIT_ENV_ARG_LIMIT=32
CONFIG_LOCALVERSION=""
CONFIG_LOCALVERSION_AUTO=y
CONFIG_SWAP=y
CONFIG_SYSVIPC=y
CONFIG_INITRAMFS_SOURCE=""
CONFIG_UID16=y
CONFIG_SYSCTL=y
CONFIG_KALLSYMS=y
CONFIG_HOTPLUG=y
CONFIG_PRINTK=y
CONFIG_BUG=y
CONFIG_ELF_CORE=y
CONFIG_BASE_FULL=y
CONFIG_FUTEX=y
CONFIG_EPOLL=y
CONFIG_SHMEM=y
CONFIG_SLAB=y
CONFIG_VM_EVENT_COUNTERS=y
CONFIG_RT_MUTEXES=y
CONFIG_BASE_SMALL=0
```

CONFIG_MODULES=y
CONFIG_MODULE_UNLOAD=y
CONFIG_MODULE_FORCE_UNLOAD=y
CONFIG_KMOD=y
CONFIG_LBD=y
CONFIG_IOSCHED_NOOP=y
CONFIG_IOSCHED_AS=y
CONFIG_IOSCHED_DEADLINE=y
CONFIG_IOSCHED_CFQ=y
CONFIG_DEFAULT_CFQ=y
CONFIG_DEFAULT_IOSCHED="cfq"
CONFIG_X86_PC=y
CONFIG_MPENTIUMIII=y
CONFIG_X86_GENERIC=y
CONFIG_X86_CMPXCHG=y
CONFIG_X86_XADD=y
CONFIG_X86_L1_CACHE_SHIFT=7
CONFIG_RWSEM_XCHGADD_ALGORITHM=y
CONFIG_GENERIC_CALIBRATE_DELAY=y
CONFIG_X86_WP_WORKS_OK=y
CONFIG_X86_INVLPG=y
CONFIG_X86_BSWAP=y
CONFIG_X86_POPAD_OK=y
CONFIG_X86_CMPXCHG64=y
CONFIG_X86_GOOD_APIC=y
CONFIG_X86_INTEL_USERCOPY=y
CONFIG_X86_USE_PPRO_CHECKSUM=y
CONFIG_X86_TSC=y
CONFIG_PREEMPT=y
CONFIG_PREEMPT_BKL=y
CONFIG_X86_MCE=y
CONFIG_VM86=y
CONFIG_NOHIGHMEM=y
CONFIG_PAGE_OFFSET=0xC0000000
CONFIG_ARCH_FLATMEM_ENABLE=y
CONFIG_ARCH_SPARSEMEM_ENABLE=y
CONFIG_ARCH_SELECT_MEMORY_MODEL=y
CONFIG_SELECT_MEMORY_MODEL=y
CONFIG_FLATMEM_MANUAL=y
CONFIG_FLATMEM=y
CONFIG_FLAT_NODE_MEM_MAP=y
CONFIG_SPARSEMEM_STATIC=y
CONFIG_SPLIT_PTLOCK_CPUS=4
CONFIG_MTRR=y
CONFIG_SECCOMP=y
CONFIG_HZ_250=y
CONFIG_HZ=250
CONFIG_PHYSICAL_START=0x100000
CONFIG_COMPAT_VDSO=y

CONFIG_PM=y
CONFIG_PM_LEGACY=y
CONFIG_ACPI=y
CONFIG_ACPI_SLEEP=y
CONFIG_ACPI_SLEEP_PROC_FS=y
CONFIG_ACPI_AC=y
CONFIG_ACPI_BATTERY=y
CONFIG_ACPI_BUTTON=y
CONFIG_ACPI_VIDEO=m
CONFIG_ACPI_FAN=y
CONFIG_ACPI_PROCESSOR=y
CONFIG_ACPI_THERMAL=y
CONFIG_ACPI_BLACKLIST_YEAR=0
CONFIG_ACPI_EC=y
CONFIG_ACPI_POWER=y
CONFIG_ACPI_SYSTEM=y
CONFIG_X86_PM_TIMER=y
CONFIG_PCI=y
CONFIG_PCI_GOANY=y
CONFIG_PCI_BIOS=y
CONFIG_PCI_DIRECT=y
CONFIG_PCI_MMCONFIG=y
CONFIG_ISA_DMA_API=y
CONFIG_ISA=y
CONFIG_BINFMT_ELF=y
CONFIG_BINFMT_AOUT=y
CONFIG_BINFMT_MISC=y
CONFIG_NET=y
CONFIG_PACKET=y
CONFIG_UNIX=y
CONFIG_XFRM=y
CONFIG_INET=y
CONFIG_IP_MULTICAST=y
CONFIG_IP_FIB_HASH=y
CONFIG_INET_XFRM_MODE_TRANSPORT=y
CONFIG_INET_XFRM_MODE_TUNNEL=y
CONFIG_INET_DIAG=y
CONFIG_INET_TCP_DIAG=y
CONFIG_TCP_CONG_BIC=y
CONFIG_NETFILTER=y
CONFIG_NETFILTER_DEBUG=y
CONFIG_NETFILTER_XTABLES=m
CONFIG_NETFILTER_XT_TARGET_CLASSIFY=m
CONFIG_NETFILTER_XT_TARGET_MARK=m
CONFIG_NETFILTER_XT_TARGET_NFQUEUE=m
CONFIG_NETFILTER_XT_MATCH_COMMENT=m
CONFIG_NETFILTER_XT_MATCH_CONNTRACK=m
CONFIG_NETFILTER_XT_MATCH_DCCP=m
CONFIG_NETFILTER_XT_MATCH_ESP=m

CONFIG_NETFILTER_XT_MATCH_HELPER=m
CONFIG_NETFILTER_XT_MATCH_LENGTH=m
CONFIG_NETFILTER_XT_MATCH_LIMIT=m
CONFIG_NETFILTER_XT_MATCH_MAC=m
CONFIG_NETFILTER_XT_MATCH_MARK=m
CONFIG_NETFILTER_XT_MATCH_POLICY=m
CONFIG_NETFILTER_XT_MATCH_MULTIPTORT=m
CONFIG_NETFILTER_XT_MATCH_PKTTYPE=m
CONFIG_NETFILTER_XT_MATCH_QUOTA=m
CONFIG_NETFILTER_XT_MATCH_REALM=m
CONFIG_NETFILTER_XT_MATCH_SCTP=m
CONFIG_NETFILTER_XT_MATCH_STATE=m
CONFIG_NETFILTER_XT_MATCH_STATISTIC=m
CONFIG_NETFILTER_XT_MATCH_STRING=m
CONFIG_NETFILTER_XT_MATCH_TCPMSS=m
CONFIG_IP_NF_CONNTRACK=m
CONFIG_IP_NF_FTP=m
CONFIG_IP_NF_IRC=m
CONFIG_IP_NF_TFTP=m
CONFIG_IP_NF_AMANDA=m
CONFIG_IP_NF_QUEUE=m
CONFIG_NET_CLS_ROUTE=y
CONFIG_STANDALONE=y
CONFIG_PREVENT_FIRMWARE_BUILD=y
CONFIG_FW_LOADER=m
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_RAM_COUNT=16
CONFIG_BLK_DEV_RAM_SIZE=4096
CONFIG_BLK_DEV_RAM_BLOCKSIZE=1024
CONFIG_BLK_DEV_INITRD=y
CONFIG_IDE=y
CONFIG_BLK_DEV_IDE=y
CONFIG_BLK_DEV_IDEDISK=y
CONFIG_IDE_GENERIC=y
CONFIG_NETDEVICES=y
CONFIG_DUMMY=m
CONFIG_NET_ETHERNET=y
CONFIG_MII=y
CONFIG_LANCE=y
CONFIG_NET_PCI=y
CONFIG_PCNET32=y
CONFIG_INPUT=y
CONFIG_INPUT_MOUSEDEV=y
CONFIG_INPUT_MOUSEDEV_PSAUX=y
CONFIG_INPUT_MOUSEDEV_SCREEN_X=1024
CONFIG_INPUT_MOUSEDEV_SCREEN_Y=768
CONFIG_INPUT_KEYBOARD=y
CONFIG_KEYBOARD_ATKBD=y
CONFIG_KEYBOARD_XTKBD=m

CONFIG_INPUT_MOUSE=y
CONFIG_MOUSE_PS2=y
CONFIG_INPUT_MISC=y
CONFIG_INPUT_PCSPKR=y
CONFIG_SERIO=y
CONFIG_SERIO_I8042=y
CONFIG_SERIO_PCIPS2=y
CONFIG_SERIO_LIBPS2=y
CONFIG_GAMEPORT=y
CONFIG_VT=y
CONFIG_VT_CONSOLE=y
CONFIG_HW_CONSOLE=y
CONFIG_SERIAL_8250=y
CONFIG_SERIAL_8250_PCI=y
CONFIG_SERIAL_8250_NR_UARTS=4
CONFIG_SERIAL_8250_RUNTIME_UARTS=4
CONFIG_SERIAL_CORE=y
CONFIG_UNIX98_PTYS=y
CONFIG_LEGACY_PTYS=y
CONFIG_LEGACY_PTY_COUNT=256
CONFIG_AGP=y
CONFIG_AGP_INTEL=y
CONFIG_VIDEO_V4L2=y
CONFIG_FIRMWARE_EDID=y
CONFIG_FB=y
CONFIG_FB_CFB_FILLRECT=y
CONFIG_FB_CFB_COPYAREA=y
CONFIG_FB_CFB_IMAGEBLIT=y
CONFIG_FB_VGA16=y
CONFIG_FB_VESA=y
CONFIG_VGA_CONSOLE=y
CONFIG_VIDEO_SELECT=y
CONFIG_DUMMY_CONSOLE=y
CONFIG_FRAMEBUFFER_CONSOLE=y
CONFIG_FONTS=y
CONFIG_FONT_8x8=y
CONFIG_FONT_8x16=y
CONFIG_FONT_6x11=y
CONFIG_FONT_PEARL_8x8=y
CONFIG_FONT_ACORN_8x8=y
CONFIG_FONT_MINI_4x6=y
CONFIG_FONT_SUN8x16=y
CONFIG_FONT_SUN12x22=y
CONFIG_LOGO=y
CONFIG_LOGO_LINUX_MONO=y
CONFIG_LOGO_LINUX_VGA16=y
CONFIG_LOGO_LINUX_CLUT224=y
CONFIG_SOUND=y
CONFIG_SND=y

CONFIG_SND_TIMER=y
CONFIG_SND_PCM=y
CONFIG_SND_RAWMIDI=y
CONFIG_SND_SUPPORT_OLD_API=y
CONFIG_SND_VERBOSE_PROCFS=y
CONFIG_SND_AC97_CODEC=y
CONFIG_SND_AC97_BUS=y
CONFIG_SND_ENS1370=y
CONFIG_SND_ENS1371=y
CONFIG_SND_INTEL8X0=y
CONFIG_SND_INTEL8X0M=y
CONFIG_USB_ARCH_HAS_HCD=y
CONFIG_USB_ARCH_HAS_OHCI=y
CONFIG_USB_ARCH_HAS_EHCI=y
CONFIG_EXT2_FS=y
CONFIG_REISERFS_FS=y
CONFIG_REISERFS_PROC_INFO=y
CONFIG_REISERFS_FS_XATTR=y
CONFIG_FS_POSIX_ACL=y
CONFIG_INOTIFY=y
CONFIG_INOTIFY_USER=y
CONFIG_DNOTIFY=y
CONFIG_AUTOFS4_FS=y
CONFIG_ISO9660_FS=y
CONFIG_JOLIET=y
CONFIG_ZISOFS=y
CONFIG_ZISOFS_FS=y
CONFIG_PROC_FS=y
CONFIG_PROC_KCORE=y
CONFIG_SYSFS=y
CONFIG_TMPFS=y
CONFIG_RAMFS=y
CONFIG_NFS_FS=y
CONFIG_NFS_V3=y
CONFIG_NFS_V3_ACL=y
CONFIG_NFS_V4=y
CONFIG_NFSD=y
CONFIG_LOCKD=y
CONFIG_LOCKD_V4=y
CONFIG_EXPORTFS=y
CONFIG_NFS_ACL_SUPPORT=y
CONFIG_NFS_COMMON=y
CONFIG_SUNRPC=y
CONFIG_SUNRPC_GSS=y
CONFIG_RPCSEC_GSS_KRB5=y
CONFIG_MSDOS_PARTITION=y
CONFIG_NLS=y
CONFIG_NLS_DEFAULT="iso8859-2"
CONFIG_NLS_CODEPAGE_437=y

CONFIG_NLS_CODEPAGE_852=y
CONFIG_NLS_CODEPAGE_1250=y
CONFIG_NLS_ISO8859_1=y
CONFIG_NLS_ISO8859_2=y
CONFIG_PROFILING=y
CONFIG_OPROFILE=y
CONFIG_TRACE_IRQFLAGS_SUPPORT=y
CONFIG_UNUSED_SYMBOLS=y
CONFIG_LOG_BUF_SHIFT=14
CONFIG_DEBUG_BUGVERBOSE=y
CONFIG_EARLY_PRINTK=y
CONFIG_DOUBLEFAULT=y
CONFIG_CRYPT=y
CONFIG_CRYPT_HMAC=y
CONFIG_CRYPT_MD5=y
CONFIG_CRYPT_SHA1=m
CONFIG_CRYPT_DES=y
CONFIG_CRYPT_DEFLATE=m
CONFIG_CRC32=y
CONFIG_ZLIB_INFLATE=y
CONFIG_ZLIB_DEFLATE=m
CONFIG_TEXTSEARCH=y
CONFIG_TEXTSEARCH_KMP=m
CONFIG_TEXTSEARCH_BM=m
CONFIG_TEXTSEARCH_FSM=m
CONFIG_PLIST=y
CONFIG_GENERIC_HARDIRQS=y
CONFIG_GENERIC_IRQ_PROBE=y
CONFIG_X86_BIOS_REBOOT=y
CONFIG_KTIME_SCALAR=y

PŘÍLOHA P IV: KONFIGURAČNÍ SOUBOR PRO PŘEKLAD NÁSTROJŮ BUSYBOX

Pro jednoduchost nejsou zobrazeny zakomentované řádky. Jde o soubor `/usr/src/busybox-1.4.1/.config`. Tento soubor lze dále editovat nejlépe pomocí příkazu `make menuconfig`, soubor pro editaci musí být umístěn v adresáři `/usr/src/busybox-1.4.1` a i spuštění příkazu musí proběhnout v tomtéž adresáři.

```
HAVE_DOT_CONFIG=y
CONFIG_SHOW_USAGE=y
CONFIG_GETOPT_LONG=y
CONFIG_FEATURE_DEVPTS=y
CONFIG_BUSYBOX_EXEC_PATH="/proc/self/exe"
CONFIG_STATIC=y
CROSS_COMPILER_PREFIX=""
CONFIG_DEBUG_YANK_SUSv2=y
CONFIG_INSTALL_APPLET_SYMLINKS=y
PREFIX="./_install"
CONFIG_MD5_SIZE_VS_SPEED=2
CONFIG_CAT=y
CONFIG_CHROOT=y
CONFIG_DF=y
CONFIG_ECHO=y
CONFIG_FEATURE_FANCY_ECHO=y
CONFIG_LS=y
CONFIG_FEATURE_LS_FILETYPES=y
CONFIG_FEATURE_LS_FOLLOWLINKS=y
CONFIG_FEATURE_LS_RECURSIVE=y
CONFIG_FEATURE_LS_SORTFILES=y
CONFIG_FEATURE_LS_TIMESTAMPS=y
CONFIG_FEATURE_LS_USERNAME=y
CONFIG_FEATURE_LS_COLOR=y
CONFIG_RM=y
CONFIG_TEST=y
CONFIG_UNAME=y
CONFIG_FEATURE_AUTOWIDTH=y
CONFIG_FEATURE_HUMAN_READABLE=y
CONFIG_START_STOP_DAEMON=y
CONFIG_FEATURE_START_STOP_DAEMON_FANCY=y
CONFIG_HALT=y
CONFIG_MESG=y
CONFIG_LSMOD=y
CONFIG_FEATURE_LSMOD_PRETTY_2_6_OUTPUT=y
CONFIG_FEATURE_CHECK_TAINTED_MODULE=y
CONFIG_DMESG=y
```

CONFIG_MOUNT=y
CONFIG_FEATURE_MOUNT_NFS=y
CONFIG_PIVOT_ROOT=y
CONFIG_UMOUNT=y
CONFIG_FEATURE_UMOUNT_ALL=y
CONFIG_FEATURE_MOUNT_LOOP=y
CONFIG_HOSTNAME=y
CONFIG_IFCONFIG=y
CONFIG_FEATURE_IFCONFIG_STATUS=y
CONFIG_FEATURE_IFCONFIG_HW=y
CONFIG_IP=y
CONFIG_FEATURE_IP_ADDRESS=y
CONFIG_FEATURE_IP_LINK=y
CONFIG_FEATURE_IP_ROUTE=y
CONFIG_APP_UDHCPC=y
CONFIG_FEATURE_SH_IS_ASH=y
CONFIG_ASH=y
CONFIG_ASH_JOB_CONTROL=y
CONFIG_ASH_ALIAS=y
CONFIG_ASH_MATH_SUPPORT=y
CONFIG_ASH_BUILTIN_ECHO=y
CONFIG_ASH_BUILTIN_TEST=y
CONFIG_ASH_MAIL=y
CONFIG_ASH_OPTIMIZE_FOR_SIZE=y
CONFIG_FEATURE_COMMAND_EDITING=y
CONFIG_FEATURE_COMMAND_HISTORY=15
CONFIG_FEATURE_IPC_SYSLOG_BUFFER_SIZE=0

PŘÍLOHA P V: SKRIPT PRO START TERMINÁLU

Pro přehlednost nejsou zobrazeny komentáře. Jde o soubor `/boot/initrd-tree/linuxrc`. Po každé editaci souboru je nutné znovu vytvořit obraz ramdisku a nebo editaci provádět s připojeným ramdiskem.

```
echo "mounting /proc filesystem"
mount -n proc /proc -t proc
echo 0x0100 > /proc/sys/kernel/real-root-dev
echo "umounting /proc filesystem"
umount /proc
echo "enabling loopback interface"
ifconfig lo 127.0.0.1 up
echo "getting IP address from DHCP server"
udhcpc --now --quit --interface=eth0 --
script=/sbin/dhcp/sample.script
echo "mounting NFS root filesystem"
mount -o nolock $ROOTDEV /mnt
echo "switching to NFS filesystem"
cd /mnt
pivot_root . old_root
echo "mounting /proc filesystem"
mount -n proc /proc -t proc
echo "starting init"
exec chroot . sh -c 'umount /old_root; exec /sbin/init'
<dev/console >dev/console2>&1
echo "exitting initrd linuxrc script"
exit 0
```