

**UNIVERZITA PARDUBICE
ÚSTAV ELEKTROTECHNIKY A
INFORMATIKY**

**System pro automatické obnovování systémů na
PC v síťové laboratoři**

BAKALÁŘSKÁ PRÁCE

2007

Ladislav Šočík

**UNIVERZITA PARDUBICE
ÚSTAV ELEKTROTECHNIKY A
INFORMATIKY**

**System pro automatické obnovování systémů na
PC v síťové laboratoři**

BAKALÁŘSKÁ PRÁCE

AUTOR PRÁCE: Ladislav Šorčík

VEDOUcí PRÁCE: Ing. Martin Dobrovolný

2007

**UNIVERSITY OF PARDUBICE
INSTITUTE OF ELECTRICAL
ENGINEERING
AND INFORMATICS**

**The System for automatic system recovery on
computers in a network laboratory**

BACHELOR WORK

AUTHOR: Ladislav Šorčík

SUPERVISOR: Ing. Martin Dobrovolný

2007

Vysokoškolský ústav: Ústav elektrotechniky a informatiky

Katedra/Ústav: Ústav elektrotechniky a informatiky

Akademický rok: 2006/2007

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Pro: Šorčík Ladislav

Studijní program: Informační technologie

Studijní obor: Informační technologie

Název tématu: Vytvořte systém pro automatické obnovování systémů na PV v síťové laboratoři

Zásady pro zpracování:

- Vytvořte systém pomocí kterého bude zautomatizováno obnovování (zálohování) diskových oblastí s OS na studentských počítačích v laboratořích. Systém umožní rychle obnovit diskové obrazy s OS v případě poškození. Systém bude schopen obnovy diskových oblastí z lokálního či vzdáleného úložiště.
- Součástí práce bude návrh a analýza možných řešení, rozbor vlastností dostupných komerčních systémů.
- Vlastní návrh systému
- Detailní popis navrhovaného systému včetně popisu zdrojových kódů
- Jednoduchá uživatelská dokumentace

Rozsah: Přibližně 30 stran

Vedoucí práce: Ing. Martin Dobrovolný

Vedoucí katedry (ústavu): prof. Ing. Pavel Bezoušek, CSc.

Datum zadání práce: 31. 10. 2006

Termín odevzdání práce: 18. 5. 2007

Prohlašuji:

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 18. 5. 2007

Ladislav Šorčík

Na tomto místě bych rád poděkoval Ing. Martinu Dobrovolnému za ochotné konzultace a za inovativní nápady. Dále pak Mgr. Tomáši Hudcovi za cenné rady, především v problematice OS Linux.

ABSTRAKT

Cílem práce je vytvořit systém umožňující snadnou obnovu systémových diskových oddílů na počítačích v síťové laboratoři. A to jak systémy na UNIXovém základě, tak systémy Windows. Systém bude umožňovat tvorbu obrazů disků a následnou obnovu systému z těchto záloh.

1	ÚVOD	10
2	VÝCHOZÍ SITUACE	11
2.1	ROZDĚLENÍ DISKŮ A INSTALOVANÉ SYSTÉMY.....	11
2.2	SOUČASNÉ OBNOVOVÁNÍ SYSTÉMU	12
2.3	POŽADAVKY NA SYSTÉM	13
3	ZPŮSOBY ŘEŠENÍ A KOMERČNÍ PRODUKTY	14
3.1	PARAGON SYSTEM RECOVERY	14
3.2	WINRESCUE XP.....	15
3.3	SYMANTEC BACKUP EXEC SYSTEM RECOVERY.....	15
3.4	NORTON GHOST.....	16
3.5	ACRONIS TRUE IMAGE	16
3.6	PARAGON DRIVE BACKUP	17
3.7	POUŽITÍ OPEN SOURCE ŘEŠENÍ	17
3.7.1	<i>Program DD</i>	17
3.7.2	<i>Partimage</i>	19
3.8	VYTVOŘENÍ VLASTNÍHO SYSTÉMU.....	21
4	NÁVRH SYSTÉMU	21
4.1	STRUKTURA	22
4.1.1	<i>Lokální úložiště</i>	22
4.1.2	<i>Síťové úložiště, NFS</i>	23
4.1.3	<i>Síťové úložiště, SMB</i>	24
4.2	POUŽITÝ SOFTWARE	25
4.2.1	<i>Základ systému</i>	25
4.2.2	<i>Klient SMB protokolu</i>	25
4.2.3	<i>DFCLDD</i>	26
4.3	VYTVOŘENÍ SYSTÉMU	26
4.3.1	<i>Instalace systému</i>	26
4.3.2	<i>Nastavení systému</i>	29
4.3.3	<i>Uživatelské rozhraní</i>	30
4.4	VYTVOŘENÍ LIVECD.....	34
4.5	ÚPRAVA LIVECD.....	37
5	ZÁVĚR	38
6	SEZNAM OBRÁZKŮ A TABULEK	39
6.1	SEZNAM OBRÁZKŮ	39
6.2	SEZNAM TABULEK	40

7	POUŽITÉ ZKRATKY A POJMY	41
8	POUŽITÉ ZDROJE.....	42
9	SEZNAM PŘÍLOH.....	43

1 Úvod

Úkolem práce je vytvořit systém umožňující rychlou a snadnou obnovu systému na počítačích v síťové laboratoři.

Daný systém bude schopen provádět obnovu jak systému na Unixovém základě (Unix, GNU/Linux, BSD), tak systému Windows. Bude tedy platformě nezávislý. Tato vlastnost je velmi důležitá, neboť v rámci výuky se studenti setkávají s mnoha druhy operačních systémů.

Na začátku práce popíše výchozí stav v síťové laboratoři, poté navrhne a analyzuje několik možných přístupů k řešení dané problematiky. Nejvhodnější řešení rozvede do funkčního systému.

Jako možná řešení také proberu několik komerčně nabízených produktů, jejich vlastností, výhod a nevýhod.

Navrhovaná řešení budou vycházet z použití GNU/Linuxového systému jako základu pro obnovovací systém.

2 Výchozí situace

V době vzniku této práce je v síťové laboratoři, kromě jiného, především síťového hardwaru, 12 počítačů určených pro výuku. Probíhá zde výuka předmětů zaměřených na získávání znalostí v oblasti hardwaru, počítačových sítí a správy operačních systémů.

Počítače tedy nejsou používány pouze pro využívání programového vybavení jako v předmětech zaměřených na získání znalostí v oblasti programovacích technik či databázových systémů, ale dochází i k manipulaci s nastavením operačního systému.

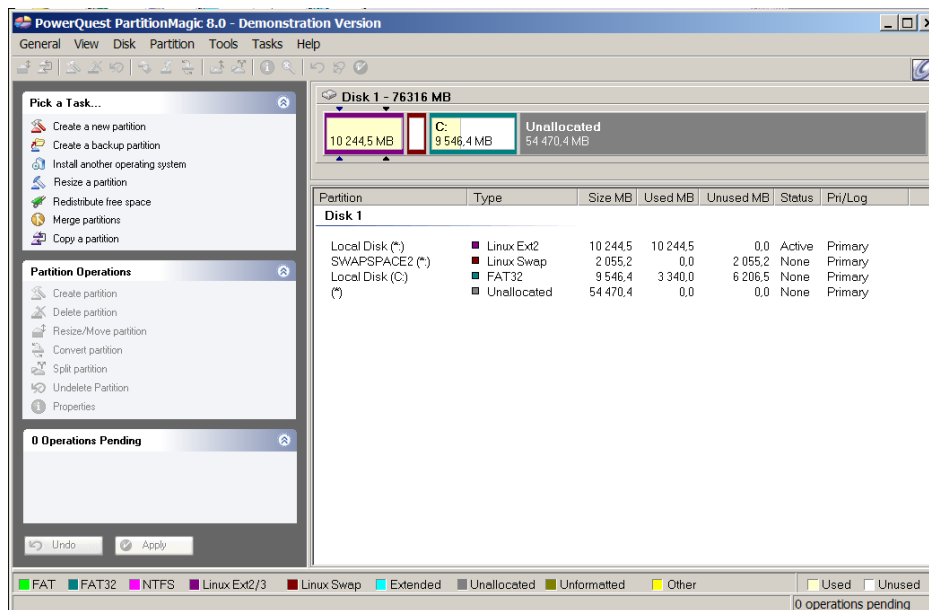
Může tedy dojít k poškození systému, a to i přes užívání softwaru pro virtualizaci (konkrétně produkt wmware). Proto je nutné mít nějaký ucelený systém umožňující rychlou a snadnou obnovu systému.

2.1 Rozdělení disků a instalované systémy

V současné době mají disky na počítačích v síťové laboratoři kapacitu 76 GB. Tato kapacita je rozdělena do 3 diskových oddílů. První má kapacitu 10 GB, druhý 2 GB a třetí 9 GB. Zbýlých 55 GB je nealokováno.

První diskový oddíl je naformátován nežurnálovým systémem souborů Ext2. Druhý diskový oddíl je naformátovaný jako odkládací (swap). Třetí je naformátován systémem souborů FAT32. První diskový oddíl je označen jako aktivní.

Na první diskový oddíl je nainstalována linuxová distribuce SuSe 10. Na oddíle naformátovaném na FAT32 je nainstalován systém Windows XP od firmy Microsoft.



Obrázek 1 - rozdělení disku na počítači

2.2 Současné obnovování systému

Obnova poškozených systémů probíhá ze zálohovaných obrazů systémových oddílů. Tyto zálohy jsou uloženy na síťovém serveru. Je k nim přístupováno pomocí NFS.

Při poškození systému je nutno nastartovat systém pomocí záchranného CD s nějakou live distribucí linuxového systému. Poté je nutné pomocí bash příkazu mount připojit přes NFS síťový disk s uloženými obrazy systémových oddílů. Obrazy jsou pak pomocí příkazu dd zkopírovány do příslušného diskového oddílu v příslušném souboru v adresáři /dev.

Příklad č. 1: současná obnova disku ze zálohy

```
#odpojení filesystemu na hda1
umount /dev/hda1

#vytvoření adresáře pro připojení úložiště
mkdir /mnt/smb

#připojení úložiště
mount -t nfs 192.168.1.1:/mnt/sdc/netlab
/mnt/smb
```

```
#překopírování obrazu disku na disk
dd if=/mnt/smb/obrazyHD/suse.img of=/dev/hda
```

Zálohy systémových oddílů se vytváří analogickou cestou jako výše zmíněná obnova systému. Na počítač se nainstaluje operační systém. Po jeho úplné konfiguraci se nastartuje systém pomocí záchranného CD s nějakou live distribucí linuxového systému. Pak se připojí pomocí příkazu mount přes NFS síťové úložiště záloh. Poté se pomocí příkazu dd překopíruje obsah systémového oddílu z příslušného souboru v adresáři /dev do úložiště.

Příklad č. 2: současná záloha disku

```
#odpojení filesystemu na hda1
umount /dev/hda1

#vytvoření adresáře pro připojení úložiště
mkdir /mnt/smb

#připojení úložiště
mount -t nfs 192.168.1.1:/mnt/sdc/netlab
/mnt/smb

#překopírování obrazu disku na disk
dd if=/dev/hda of=/mnt/smb/obrazyHD/suse.img
```

2.3 Požadavky na systém

Systém by měl být schopný zautomatizovat provádění úkonů při zálohování systémových diskových oddílů, umožňovat zálohu jednotlivých diskových oddílů i celého disku a zprostředkovávat jednoduché uživatelské rozhraní umožňující provádět zálohování i uživatelům, kteří neznají systém GNU/Linux.

Dále by měl podporovat základní možnosti komprimace obrazů disků, aby docházelo k šetření systémových prostředků serverů (především diskových) v síťové laboratoři.

3 Způsoby řešení a komerční produkty

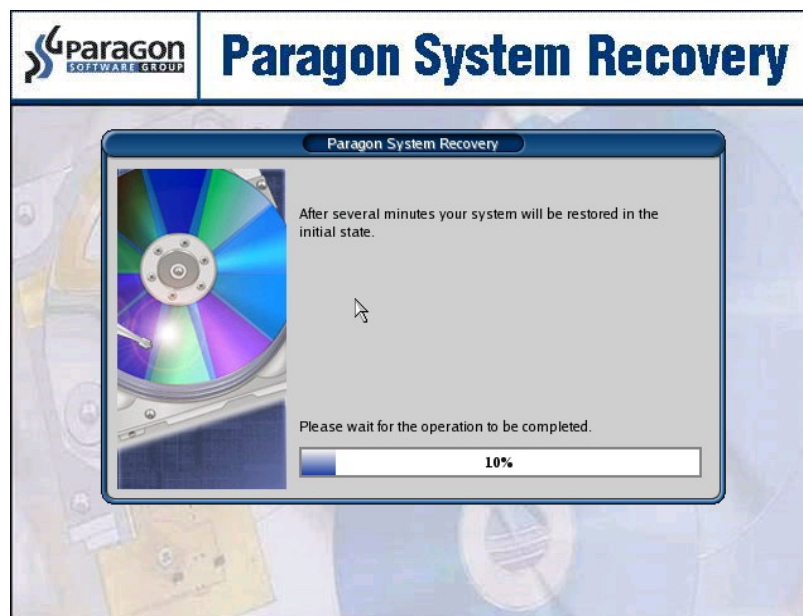
V současné době se na trhu vyskytuje několik produktů umožňujících zálohování systémových oddílů. Jejich neflexibilita a orientace především na platformu Microsoft Windows v podstatě znemožňuje jejich nasazení v síťové laboratoři.

3.1 Paragon system recovery

Řešení od firmy Paragon nabízí východisko především pro výrobce hardwaru. Obrazy síťových disků mohou být uloženy na zvláštním CD/DVD nebo ve zvláštním oddílu na disku nazvaném „Backup capsule“. Toto řešení tedy neumožňuje ukládání obrazů přes síť přímo na server.

Prostředí obnovovacího softwaru je plně modifikovatelné podle požadavku zákazníka, lze modifikovat loga, popisky atp. Dále je možné zvolit na jakém základě bude systém fungovat, zda-li systém DOS či GNU/Linux.

Daný systém není vhodný pro nasazení v síťové laboratoři především pro nemožnost provádět zálohu přes síť.



Obrázek 2 - Paragon System Recovery

3.2 WinRescue XP

Programová sada od firmy Super Win Software, Inc. je určena výhradně pro systém Windows XP. Je tedy pro použití v síťové laboratoři naprosto nevhodná.

Umožňuje sice velké množství funkcí – od zálohování jednotlivých souborů, přes odstraňování hardwarových problémů až po obnovení systému do tzv. WinXP Restore Points.

3.3 Symantec Backup Exec System recovery

Produkt od firmy Symantec Corp., která mimo jiné vyvíjí i program Ghost, je opět určen jen pro systém Windows. Jeho použití v síťové laboratoři tedy nepřipadá v úvahu.

Původně byl vyvinut jako nástroj umožňující snadné a rychlé instalování systému na identické hardwarové sestavy, postupně se začal používat také pro zálohování systému.

Umožňuje zálohování téměř na všechna úložiště včetně síťových, vyjma zálohovacích pásků. Automatické zálohování systému umožňuje pravidelné uložení stavu systému.

K základnímu obrazu disku se zálohují postupné změny v systému, před které je možné se při obnově vrátit.

3.4 Norton Ghost

Program produkováný společností Symantec Corp. byl prvním komerčně úspěšným nástrojem pro klonování diskových oddílů. Během jeho vývoje došlo ke sloučení s jeho největším konkurentem DriveImage od společnosti PowerQuest. Od verze 9 je Ghost založen na jádře DriveImage, není tedy přímo zpětně kompatibilní s obrazy vytvořenými v předchozích verzích. Kompatibilitu lze docílit pomocí volby „restore legacy image“.

Nativně Ghost podporuje, kromě souborových systémů určených pro Windows (FAT32 a NTFS), také souborové systémy Ext2 a Ext3. Lze ho tedy využít i pro zálohování linuxových systémových oddílů. V případě zakoupení licencí by tedy bylo možné jeho použití v síťové laboratoři. Velkou nevýhodou je však fakt, že kromě Ext2 a Ext3 nepodporuje další linuxové souborové systémy, např. reiserfs.

Poslední verze již obsahují podporu pro Windows Vista a GUID Partition table[1].

3.5 Acronis True Image

Nástroj od společnosti Acronis je další z řady nástrojů určených pro Windows. Jelikož neposkytuje možnost zálohování newindowsovských souborových systémů, je jeho použití v síťové laboratoři nemožné.

Umožňuje ukládání jak obrazu celého disku, tak zálohu jednotlivých složek a souborů přesně podle požadavků uživatele. Lze tedy zálohovat např. pouze dokumenty, hudbu, nebo poštu z e-mailového klienta.

3.6 Paragon drive backup

Další nástroj od společnosti Paragon představuje komplexní řešení pro zálohování a správu diskových oddílů, celých disků a jejich obrazů.

Prostředí nástroje lze spustit jen ve Windows (nově také ve Windows Vista).

Nástroj umožňuje zálohování jednotlivých oddílů, sdruženou zálohu několika oddílů, nebo zálohu celého disku. Dále lze zazálohovat master boot record, periodické zálohování a udržování obrazů aktuálních. Dále je možné provádět přímé kopírování disků (vytvářet identické disky). Pomocí nástroje Volume explorer je možné z obrazu obnovit jen určité soubory, nebo složky.

Obrazy lze ukládat na místní disk, na zálohovací zařízení připojené přes USB či firewire, do síťového úložiště, do speciálního skrytého diskového oddílu, tzv. „Backup capsule“, nebo do ISO obrazu pro pozdější vypálení obrazu na DVD-R či CD-R.

Obrazy vytvořené tímto nástrojem lze přímo archivovat a efektivně tak zmenšovat diskové místo potřebné pro jejich uchování. Pro větší přehlednost je možné přidávat k archivům popisky.

3.7 Použití open source řešení

3.7.1 Program DD

Program *dd* je běžný unixový program určený pro nízkourovňové kopírování souborů. Název *dd* je zkratka od slov „data definition“.

Program *dd* obsahuje řadu rozdílných příkazů oproti dalšímu unixovému nástroji *cp*. *Cp* kopíruje soubory celé, zatímco

dd umožňuje kopírování dat po blocích předem dané délky s podporou konverze kódování během kopírování.

Program *dd* vytváří na bajt přesnou kopii originálního souboru, proto bývá často využíván jako nástroj pro zálohování diskových oddílů či bootovacích sektorů disků. Za tímto účelem je také používán při současném zálohování systémových oddílů.

V rámci operačních systémů Unix a GNU/Linux je vše, co nějakým způsobem komunikuje s operačním systémem (od zařízení až po procesy), chápáno jako soubor v rámci stromu systému souborů. Program *dd* má tedy téměř neomezené možnosti. Lze ho například použít k výstupu na externí zařízení atp.

Díky možnosti *rour*, lze výstup z programu *dd* napojit na vstup komprimačního nástroje a naopak. Lze tedy při zálohování či výstupu ze zařízení ukládat data v komprimované podobě.

Syntaxe programu *dd* je značně odlišná od syntaxe běžných linuxových programů a nástrojů. Následuje výčet základních přepínačů programu.

- *if=soubor* – specifikuje vstupní soubor
- *of=soubor* – specifikuje výstupní soubor
- *ibs=bajty* – specifikuje velikost bloku na vstupu
- *obs=bajty* – specifikuje velikost bloku na výstupu
- *bs=bajty* – specifikuje velikost bloku (na vstupu i na výstupu), pokud není zadána hodnota, použije se výchozí hodnota 512 bajtů
- *count=bloky* – specifikuje počet bloků ke zkopírování
- *skip=bloky* – specifikuje počet bloků, které mají být přeskočeny na začátku čtení

- seek=*bloky* – specifikuje počet bloků, které mají být přeskočeny při zápisu (zápis nebude začínat od začátku cílového souboru)

3.7.2 Partimage

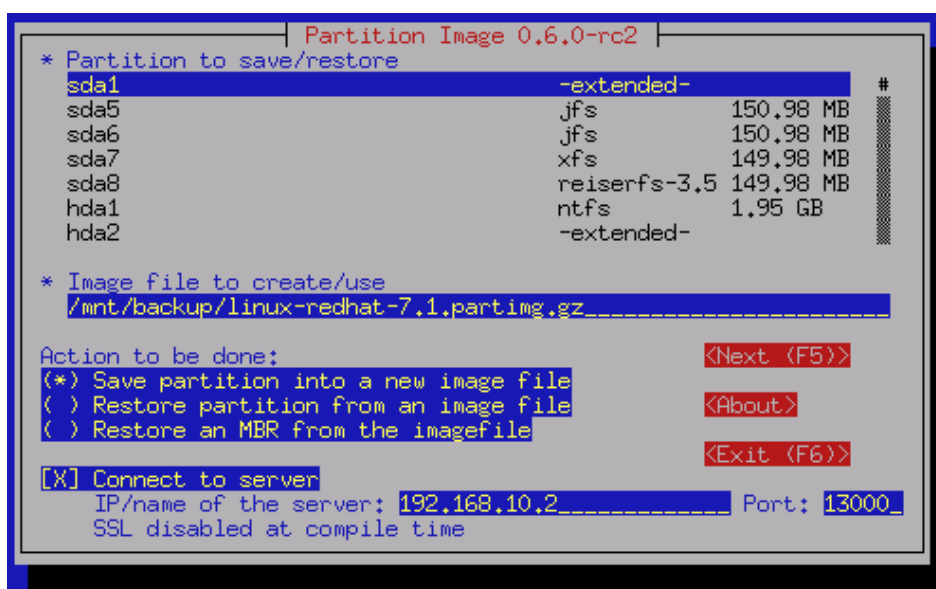
Partimage je linuxový nástroj určený pro zálohu systémových oddílů. Podporuje většinu hlavních souborových systémů.

Tabulka 1 – Podpora souborových systémů v programu Partimage

Název	Popis	Podpora
Ext2/Ext3	standartní linuxový systém souborů	stabilní
Reiser3	žurnálový a výkonný systém souborů	stabilní
FAT16/32	systém souborů používaný operačními systémy DOS a Windows	stabilní
HPFS	systém souborů používaný operačním systémem OS/2 od firmy IBM	stabilní
JFS	žurnálový souborový systém vyvinutý firmou IBM	stabilní
XFS	žurnálový a úsporný systém souborů	stabilní
UFS	unixový souborový systém	beta
HFS	systém souborů používaný operačním systémem MacOS	beta
NTFS	systém souborů používaný operačními systémy Windows NT, Windows XP a Windows Vista	testovací

Partimage je vhodné řešení pro zálohování diskových oddílů používajících linuxové a unixové souborové systémy. Obrazy disků vytvořené pomocí Partimage lze použít k obnovení systému v případě poškození nebo pro hromadnou instalaci většího počtu stanic stejné hardwarové konfigurace.

Partimage ukládá do obrazů diskových oddílů pouze data. Na rozdíl od programu *dd*, který kopíruje veškeré bajty souborového systému, tedy i prázdné bajty neobsazené daty. Tím dochází k úspoře místa na úložisti. K další úspoře místa dochází podporou komprese. Ke komprimování je využíván GZIP/BZIP2 formát.



Obrázek 3 - Partimage

Program Partimage obsahuje jen základní podporu souborového systému NTFS používaného operačními systémy Windows.

Diskový oddíl se systémem souborů NTFS půjde pomocí Partimage zazálohovat a následně obnovit pouze v případě, že data na diskovém oddílu nebudou příliš fragmentovaná. Chyba při zálohování diskového oddílu se projeví také v případě, že systémové soubory na daném diskovém oddílu jsou komprimované.

3.8 Vytvoření vlastního systému

Vytvoření vlastního systému přinese především značné zjednodušení procesu obnovy systému, především pro umožnění ovládní systému i uživateli bez hlubších znalostí operačního systému GNU/Linux.

Vlastní systém bude oproti výše zmíněným produktům vytvořen přesně podle potřeb síťové laboratoře. Nebude zaměřen jen na platformu Microsoft Windows. Bude umožňovat ukládání obrazů přes síť v komprimované podobě.

Vlastní systém bude vycházet z linuxové distribuce Archlinux. K plnění úkolů bude používat dříve zmíněný open source nástroj *dd* resp. jeho modifikaci *dfcldd*.

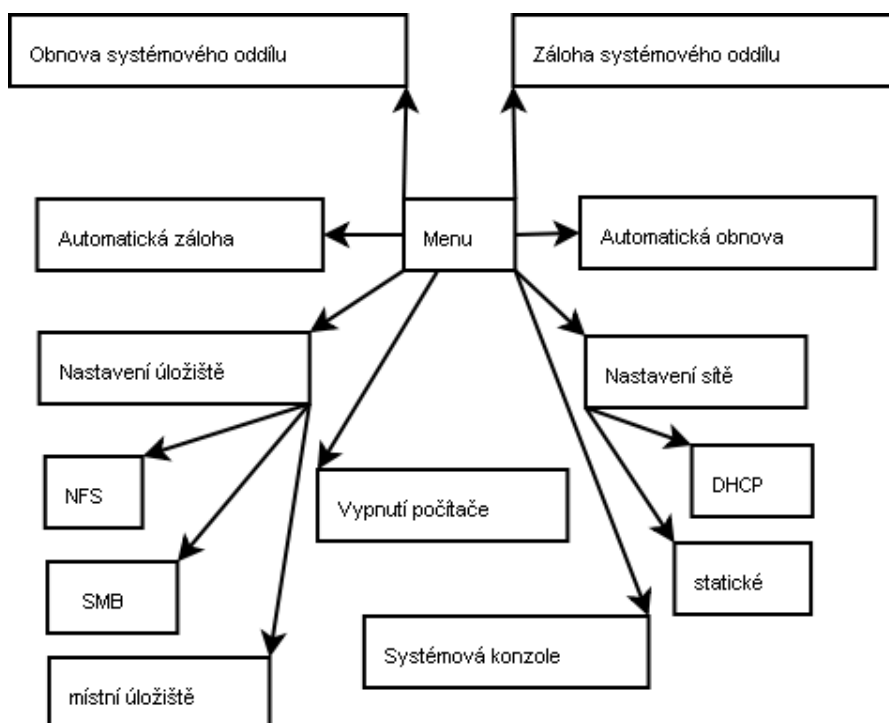
4 Návrh systému

Systém bude poskytovat uživatelské rozhraní umožňující základní funkce pro obnovu nebo zálohování. Uživatelské rozhraní bude v podobě textového menu.

Z hlavního menu bude přístup do jednotlivých submenu, ve kterých bude možno provádět jednotlivé požadované funkce.

V případě automatické volby (obnovy či zálohy) dojde k načtení parametrů z diskety. Pokud toto načtení selže, dojde k načtení implicitních parametrů. Při manuální volbě jsou parametry zadány z klávesnice.

Z uživatelského rozhraní bude také možné nastavit síťové rozhraní a typ uložení. Z uživatelského rozhraní je také možné se přepnout do systémové konzole.



Obrázek 4 - struktura uživatelského rozhraní

4.1 Struktura

System bude pracovat jako tzv. LiveCD[2]. Po nastartování systému z tohoto počítače dojde ke spuštění obslužného shell skriptu, který formou jednoduchého textového menu nabídne zálohování jednotlivých diskových oddílů do úložiště.

Souborový systém úložiště bude připojen do adresáře /storage. System bude podporovat připojení úložiště z místního disku (implicitní nastavení), přes síť pomocí protokolu NFS a přes síť pomocí protokolu Samba.

4.1.1 Lokální úložiště

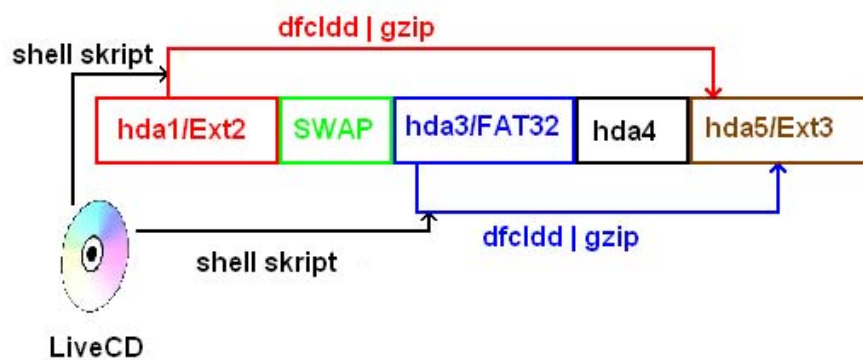
Nastavení úložiště na lokální disk je pro system implicitní. Po spuštění systému dojde tedy k připojení místního diskového oddílu (zařízení /dev/hda5) do adresáře /storage.

Po připojení diskového oddílu do adresáře /storage je staré jádro systému podle nastavení v souboru /etc/fstab.

Příklad č. 3: záznam v /etc/fstab pro připojení /storage

```
/dev/hda5 /storage ext3 defaults 0 0
```

Při zálohování dojde k překopírování obsahu příslušného souboru v adresáři /dev do zadaného souboru v adresáři /storage, jak to znázorňuje následující obrázek.

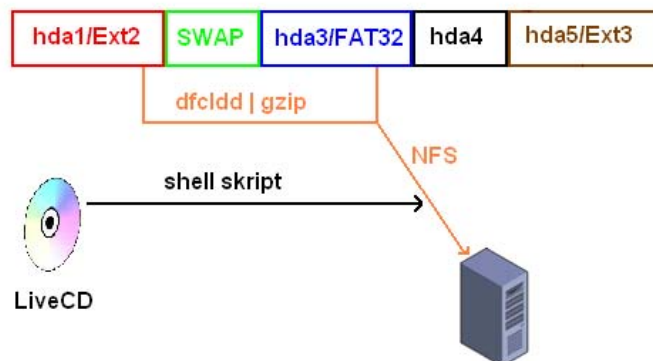


Obrázek 5 - záloha na lokální úložiště

4.1.2 Síťové úložiště, NFS

Pomocí volby „Připojit úložiště v menu“ lze zvolit připojení souborového systému pomocí protokolu NFS do adresáře /storage. Po zadání adresy serveru dojde k pokusu o připojení zadaného adresáře na serveru do adresáře /storage.

Zálohování poté probíhá obdobně jako při použití lokální úložiště. Data jsou však posílána přes síť na server.

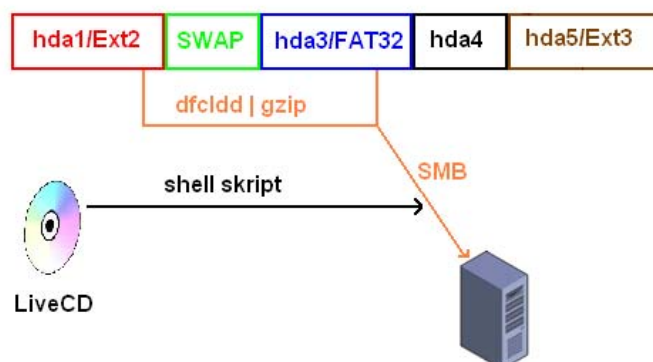


Obrázek 6 - záloha na síťové úložiště, NFS

4.1.3 Síťové úložiště, SMB

Připojení síťového úložiště přes protokol SMB je analogické k připojení úložiště přes protokol NFS. V systému je však potřeba mít nainstalovaný balík smbclient.

Po zadání adresy samba serveru je uživatel dotázán na heslo. Po jeho zadání dojde k pokusu o připojení sdílené složky storage\$ do adresáře /storage. Následné zálohování probíhá stejně jako při použití síťového úložiště přes protokol NFS.



Obrázek 7 - záloha na síťové úložiště, SMB

4.2 Použitý software

4.2.1 Základ systému

Pro obnovovací systém jsem zvolil distribuci ArchLinux. Důvodem byla především jednoduchost. V rámci této distribuce je distribuován nejnovější software. Díky jednoduché struktuře je velmi snadné ArchLinux přizpůsobit specifickým potřebám.

Aby bylo možné systém upravit pro LiveCD musí být v jádře zavedeny moduly aufs a squashfs.

4.2.2 Klient SMB protokolu

Jako klient pro připojení síťového úložiště přes protokol SMB je použi balík smbclient.

SMB (Server Message Block) byl původně vyvinut firmou IBM, nejrozšířenější verze však pochází od firmy Microsoft. SMB byl původně navržen, aby fungoval nad protokolem NetBIOS, který běží nad protokoly rodin NetBEUI, IPX/SPX či TCP/IP. SMB však může běžet přímo nad TCP/IP od Windows 2000.

V roce 1996 firma Microsoft přejmenovala protokol SMB na Common Internet File System (CIFS) a přidala další vlastnosti. Mimo jiné podporu symbolických a tvrdých odkazů, zvětšila se maximální velikost souboru. Microsoft také vydal část specifikace jako informativní RFC.

Vzhledem k důležitosti protokolu SMB v rámci spolupráce s dominantní platformou Microsoft Windows byl založen projekt Samba, který měl pomocí zpětné analýzy paketů vytvořit volně dostupného klienta pro ostatní operační systémy.

Spolu s Windows Vista představil Microsoft také novou verzi protokolu SMB 2.0. Tato verze přináší zlepšení především v oblasti bezpečnosti a výkonu.

4.2.3 DFCLDD

Jedná se o vylepšenou verzi GNU programu *dd* z balíku *Coreutils*. Oproti původní verzi přináší několik významných vylepšení. Těmi jsou :

- hashování „za letu“ – *dfcldd* umožňuje provádět hashování dat během transferu a tím zajišťuje jejich integritu,
- zobrazení průběhu kopírování – průběh kopírování lze sledovat jako počet přenesených bloků, procentuální vyjádření či jako přibližný čas zbývající do konce kopírování,
- ověřování – *dfcldd* umožňuje provést kontrolu, zda je cílová jednotka přesnou kopií vstupního souboru, či matice,
- několikanásobný výstup – *dfcldd* podporuje výstup do několika souborů či zařízení současně
- dělený výstup – *dfcldd* umožňuje dělit výstup po blocích do několika souborů

4.3 Vytvoření systému

Pro vytvoření systému jsem použil software VMWare Workstation ve zkušební verzi.

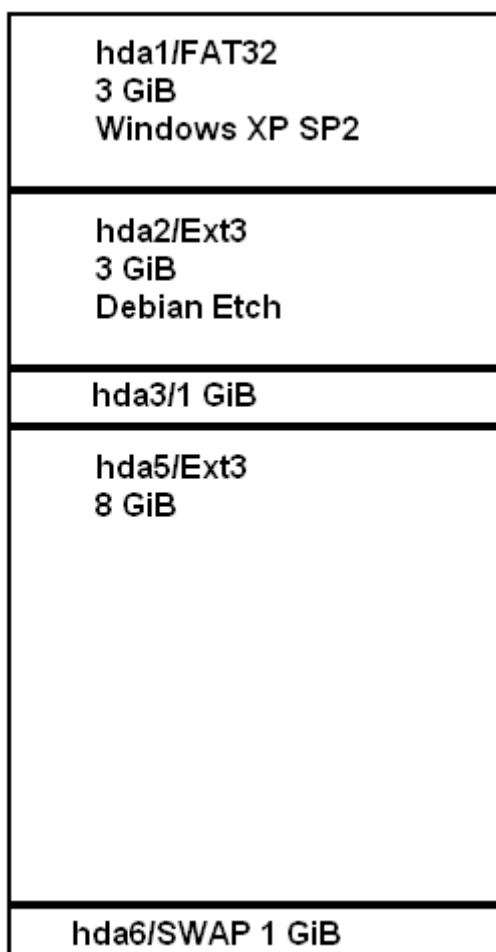
4.3.1 Instalace systému

Z oficiálních stránek linuxové distribuce ArchLinux jsem si stáhl obraz instalačního CD verze 0.8. Obraz jsem připojil v programu VMWare jako virtuální CD mechaniku a nastartoval instalaci.

Po nastartování systému z CD a naběhnutí systémové konzole jsem příkazem */arch/setup* spustil instalaci systému. Jako zdroj souboru pro instalaci jsem zvolil CD-ROM či jiný zdroj.

V prvním kroku jsem přiřivil pevný disk virtuálního počítače pro nainstalování systému.

Pevný disk jsem připravoval manuálně. Na virtuálním pevném disku byli v době instalace již diskové oddíly hda1 (se souborovým systémem FAT32 a nainstalovaným systémem Windows XP Service Pack 2, velikost 3 GiB, primární diskový oddíl), hda2 (se souborovým systémem Ext3 a nainstalovaným systémem Debian Etch, velikost 3 GiB, primární diskový oddíl), hda3 (diskový oddíl o velikosti 1 GiB připravený pro instalaci systému, primární oddíl), hda5 (s žurnálovým souborovým systémem Ext3 používaný jako úložiště dat, velikost 8 GiB) a hda6 (odkládací linuxový oddíl, tzv. „swap“, velikost 1 GiB).



Obrázek 8 - rozdělení diskových oddílů ve virtuálním PC

Jako odkládací disk systému jsem použil stejný diskový oddíl jako pro systém Debian nainstalovaný na virtuálním počítači, tedy oddíl hda6.

Jako kořenový souborový systém, jsem použil systém souborů na diskovém oddíle hda3. Tento systém souborů jsem při instalaci nechal vytvořit. Jako souborový systém jsem použil nežurnálový souborový systém Ext2.

Další diskové oddíly jsem ponechal beze změn a k nově instalovanému systému jsem je nepřipojoval. Jediný další používaný oddíl bude hda5. Jeho připojení jsem provedl později pomocí úprav v souboru */etc/fstab*.

Po vytvoření souborových systémů a připojení diskových oddílů jsem přešel k výběru balíčků.

Instalační CD obsahuje pouze balíčky ze skupiny *base*. Z této skupiny jsem neinstaloval tyto balíčky:

- bin86 – nástroj pro práci s binárními soubory pro platformu 8086,
- bzip2 – komprimační nástroj,
- lilo – zavaděč systému,
- mailx – nástroj pro posílání e-mailových zpráv z příkazové řádky,
- nano – textový editor,
- wireless_tools – nástroje pro práci s bezdrátovými zařízeními,
- wpa_supplicant – nástroj zajišťující výměnu klíčů, během wpa šifrování v bezdrátovém provozu.

Po nainstalování balíčků a vytvoření stromu jejich vzájemných závislostí jsem přešel k nastavení systému.

Povolil jsem podporu pro startování systému přes USB rozhraní pro případ, že by se pozdější systém přenášel na USB zařízení.

Nastavení systému probíhá pomocí editace konfiguračních souborů. Tyto změny se nachází v příloze C příklad č. 1 a č. 2.

Účet superuživatele (*root*) jsem ponechal bez hesla. Vzhledem k povaze systému je heslo zbytečné.

4.3.2 Nastavení systému

Po nainstalování a nastartování systému jsem musel doinstalovat balíčky ze skupiny EXTRA. Instalované balíčky byly:

- *smbclient* – nástroj umožňující připojit souborový systém přes protokol SMB,
- *nfs-utils* – sada nástrojů pro práci s protokolem NFS,
- *dcfldd* – vylepšená verze GNU programu *dd*.

Postup doinstalování balíčků se nachází v příloze C příklad č. 3.

Pro správné fungování balíku *smbclient* bylo nutné vytvořit prázdný soubor */etc/samba/smb.conf*. Pokud tento soubor neexistuje, tak pokus o připojení souborového systému vypíše chybové hlášení, ačkoliv připojení funguje.

Dále jsem doinstaloval balíčky *mc*, *eject*, *iftop* a *openssh*. Jedná se o souborový manažer podobný Volkov Commanderu či Norton Commanderu z platformy DOS/Windows a o nástroj umožňující sledování sítě. Balíček *eject* poskytuje nástroj pro otevření CD mechaniky při ukončení systému a balíček pro připojení ke vzdálenému počítači pomocí protokolu Secure Shell. Tyto balíčky nejsou vitálně důležité, ale jedná se o užitečné nástroje pro práci se systémem. Postup doinstalování balíčků se nachází v příloze C příklad č. 4.

Dále bylo nutné vytvořit adresář */storage*. Do tohoto adresáře rozhraní připojuje různá úložiště. Po vytvoření tohoto adresáře se tak záznam v souboru */etc/fstab* stane platným.

Poté jsem do domácího adresáře superuživatele *root* nahrál skript zajišťující uživatelské rozhraní systému pro obnovu. Tento skript se jmenuje *~/sys_rec.sh*. Aby se tento skript spouštěl po přihlášení superuživatele do systému, bylo nutné upravit soubor */root/.bashrc*. Úprava se nachází v příloze C příklad č. 5.

Pro automatické přihlášení superuživatele po startu systému bylo nutné vytvořit program v jazyce C, který toto přihlášení vykoná. Po jeho přeložení pomocí GNU kompilátoru *gcc*, jsem výsledný binární soubor překopíroval do adresáře */usr/local/sbin*. Zdrojový kód programu se nachází v příloze C příklad č. 6.

Aby k automatickému přihlášení superuživatele po startu k systému došlo, bylo ještě nutné upravit soubor */etc/inittab*. Tato úprava se nachází v příloze C příklad č. 7.

Aby skript spuštěný při startu mohl správně pracovat, je nutné, aby běžel s právy superuživatele. Bylo tedy nutné mu pomocí příkazu *chmod* nastavit UID bit. Soubor s nastaveným UID bitem je poté spuštěn s právy vlastníka souboru, nikoliv uživatele, který ho spustil. Příkaz, který to zajistí, je v příloze C příklad č. 8.

Aby bylo možné se do rozhraní dostat ze systémové konzole jednoduchým příkazem *menu.*, bylo nutné do souboru */root/.bashrc* přidat alias. Tyto aliasy se nachází v příloze C příklad č. 9.

4.3.3 Uživatelské rozhraní

Uživatelské rozhraní je tvořeno shellovským skriptem, který získává od uživatele údaje a provádí potřebné příkazy s parametry podle údajů od uživatele. Kompletní zdrojový kód je v příloze A.

Samotný skript obsahuje následující funkce:

- menu(),
- recovery(),
- backup(),
- mount_storage(),
- set_network().

Funkce menu() po svém zavolání vypíše hlavičku, krátkou nápovědu, momentálně připojené úložiště a nabídku možností, co lze udělat.

```
Nastaveni site (eth0):
eth0      Link encap:Ethernet HWaddr 08:0C:29:3D:E4:F1
          inet addr:10.106.10.7 Bcast:10.106.10.31 Mask:255.255.255.224
          inet6 addr: fe80::20c:29ff:fe3d:e4f1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:168 (168.0 b)
          Interrupt:18 Base address:0x1000

Hint --
volbou 3 se dostanete do konzole systemu. Login == root, heslo == rootroot
Prikazem menu z konzole se dostanete zpět do menu.

STORAGE : local
1 - Obnovit systemovy oddil
2 - Zalohovat systemovy oddil
3 - Pripojit uloziste
4 - Nastaveni site
0 - Systemova konzole
z - automaticka zaloha
o - automaticka obnova
k - Konec
Volba : _
```

Obrázek 9 - úvodní nabídka systému

Funkce recovery() po svém zavolání získá od uživatele informace o názvu souboru s obrazem uloženém v úložišti a o cílovém diskovém oddíle, kam má být obraz obnoven. Tyto informace pak předá programu *gzip* resp. *dcfldd* jako parametry.

Příklad č. 4: obnova z obrazu Windows.img.gz na diskový oddíl hda1.

```
gzip -dc /storage/Windows.img.gz | dcfldd
of=/dev/hda1 bs=8192 sizeprobe=of;
```

Obraz je nejprve pomocí programu *gzip* dekomprimován. Přepínač *d* určuje, že se má dekomprimovat. Přepínač *c* určuje, že se má výstup z programu přeměřovat na standartní výstup. Ten je pak pomocí příkazu `|` přeměřován na vstup programu *dcfldd*, který data po blocích zapíše na diskový oddíl *hda1*.

Parametr *bs* z příkladu č. 10 určuje jak velký má být jeden blok kopírovaných dat. Hodnotu 8192 jsem zvolil, aby zobrazovaný průběh kopírování byl plynulý.

Parametr *sizeprobe* z příkladu č. 10 určuje velikost souboru (vstupního či výstupního), který má být použit k určení procentuálního vyjádření průběhu kopírování.

Funkce `backup()` pracuje analogicky k funkci `back()`. Vypíše detekované diskové oddíly a od uživatele získá informace o diskovém oddíle, který má být zálohován, a názvu obrazu, který má být v úložišti vytvořen, popřípadě přepsán. Tyto informace poté předá programu *dcfldd* resp. *gzip* jako parametry.

```
Detekovane diskove oddily :
brw-rw---- 1 root disk      3, 0 2007-05-16 22:23 /dev/hda
brw-rw---- 1 root disk      3, 1 2007-05-16 22:23 /dev/hda1
brw-rw---- 1 root disk      3, 2 2007-05-16 22:23 /dev/hda2
brw-rw---- 1 root disk      3, 3 2007-05-16 22:23 /dev/hda3
brw-rw---- 1 root disk      3, 4 2007-05-16 22:23 /dev/hda4
brw-rw---- 1 root disk      3, 5 2007-05-16 22:23 /dev/hda5
brw-rw---- 1 root disk      3, 6 2007-05-16 22:23 /dev/hda6
brw-rw---- 1 root optical 22, 0 2007-05-16 22:23 /dev/hdc
0 - navrat do menu
Zadej partition k zalohovani(napr. hda1) : hda1
Zadej nazev ciloveho obrazu : WindowsXP.img.gz
Zalohuji systemovy oddil /dev/hda1 do souboru /storage/WindowsXP.img.gz ...
Mejte chvili strpeni ...
[15% of 2863Mbl 56064 blocks (438Mb) written. 00:13:56 remaining._
```

Obrázek 10 - průběh zálohování diskového oddílu *hda1* do obrazu *WindowsXP.img.gz* v úložišti

*Příklad č. 5: Zálohování diskového oddílu *hda1* do obrazu *WindowsXP.img.gz* v úložišti.*

```
dcflff if=/dev/hda1 bs=8192 sizeprobe=if
| gzip > /storage/WindowsXP.img.gz
```


Parametry *bs* a *sizeprobe* mají stejný význam jako u funkce *restore()*. Výstup programu *dcfldd* je pomocí příkazu `|` přeměřován na vstup programu *gzip*, který data zkomprimuje do cílového obrazu.

Funkce *mount_storage()* po svém zavolání podle předaného parametru získá od uživatele potřebné údaje o úložišti, které má být připojeno. Podle zadaných údajů připojí úložiště do adresáře */storage*.

Funkci je možné předat tyto hodnoty parametru:

- 1 – jako úložiště se připojí místní diskový oddíl. Tato volba je implicitní, díky záznamu v souboru */etc/fstab* je po startu systému do adresáře */storage* připojen místní diskový oddíl *hda5* se systémem souborů Ext3. Pokud se tento diskový oddíl na počítači nevyskytuje, je nutné před zálohováním připojit jiné úložiště.
- 2 – Pomocí této hodnoty je možné připojit úložiště přes síťový protokol NFS.
- 3 – Pomocí této hodnoty je možné připojit úložiště přes síťový protokol SMB.

Funkce *set_network()* po svém zavolání umožňuje upravit nastavení síťových rozhraní detekovaných v systému. Poté se dotáže uživatele, zda-li má být nastavení statické či dynamicky získané prostřednictvím protokolu DHCP[3].

V případě statického nastavení od uživatele získá informace o síťových parametrech a ty pak nastaví zvolenému rozhraní. Příklad statického nastavení sítě se nachází v příloze C příklady č. 10 a č. 11.

Skript dále obsahuje nekonečnou smyčku volající funkci `menu()`. Po získání volby od uživatele dojde k zavolání vyžadované funkce, přepnutí do konzole či zastavení systému.

V případě zvolení automatické obnovy/zálohy systém připojí disketovou mechaniku do adresáře `/mnt/floppy`. Odtud načte nastavení proměnných ze souboru `autoconf.sh`. Tyto proměnné jsou použity jako parametry pro obnovení či zálohování systému. Vzorový soubor `autoconf.sh` se nachází v obrazu live CD v adresáři `system`. V případě, že se nepodaří připojit disketovou mechaniku, použijí se hodnoty ze vzorového souboru.

4.4 Vytvoření LiveCD

Vytvořený a připravený systém bylo nutné převést na LiveCD, aby bylo možno z CD startovat systém pro zálohování.

K vytvoření obrazu LiveCD bylo potřeba do systému doinstalovat následující balíky:

- `cdrkit`
- `e2fsprogs`

Příkazy pro nainstalování balíků se nachází v příloze C příklad č. 13.

Na diskovém oddílu s dostatečnou kapacitou jsem vytvořil adresář `System_recovery` pro vytvoření adresářové struktury pro live CD. V tomto adresáři jsem vytvořil dva podadresáře `isolinux` a `system`. Z live distribuce Insert jsem zkopíroval program `isolinux.bin`, nutný pro vytvoření ISO9660 souborového systému, spolu se souborem `boot.cat` do adresáře `isolinux`.

Dále jsem vytvořil soubor s textem uvodní zprávy live CD `boot.msg`. Jako další jsem vytvořil soubor s nastavením zavaděče live CD `isolinux.cfg` (příloha C příklad č. 14).

Bylo nutné vytvořit obraz souborového systému, který se bude při startu systému načítat do operační paměti. Vytvořil jsem soubor *miniroot* a v něm jsem pomocí příkazu *mkfs.ext2* vytvořil souborový systém Ext2 o velikosti 100000 KiB (příloha C příklad č. 15).

Soubor s obrazem jsem pomocí příkazu *mount* s přepínačem *-o loop* připojil k systému do adresáře */mnt/image*.

Příklad č. 6: Připojení obrazu k systému

```
#mount -o loop miniroot /mnt/image
```

Z obrazu jsem odstranil adresář *lost+found*. Do obrazu jsem nakopíroval systémové adresáře */bin*, */etc*, */root*, */sbin*, */tmp* a */var*. Dále jsem vytvořil adresář */storage* pro připojení úložiště.

Do adresáře *system* v adresářové struktuře CD jsem nakopíroval systémové adresáře */lib*, */opt* a */usr*. V obraze jsem dovytvořil adresáře */dev*, */mnt/cdrom*, */mnt/floppy*, */mnt/dvd* a */proc*. Vytvořil jsem symbolické odkazy *lib*, *opt* a *usr* odkazující na příslušné adresáře připojené k systému na CD do adresáře */mnt/cdrom/system* (příloha C příklad č. 16).

Nástroj pro automatickou detekci hardwaru a zavedení příslušných modulů do jádra *hwdetect* potřebuje pro svoji správnou funkci program *find*. Tento program se nachází v adresáři */usr/bin*. Adresář */usr* je však připojován pomocí symbolického odkazu ze souborového systému CD. CD-ROM mechanika je však detekována právě nástrojem *hwdetect*.

Proto bylo nutné ručně zavést do jádra moduly pro ide zařízení. Tyto moduly jsou definovány polem *MODULES* v souboru */etc/rc.conf*.

Příklad č. 7: Pole modulů pro zavedení do jádra v souboru /etc/rc.conf

```
MODULES=(!usbserial ide-core ide-disk ide-cd  
ide-generic)
```

Dále jsem v obrazu v souboru */etc/fstab* (příloha C příklad č. 17) upravil záznamy, aby probíhalo připojování jednotlivých souborových systémů správně.

První záznam v souboru slouží k připojení kořenového systému. Třetí záznam určuje připojení CD mechaniky, které je důležité pro přístup k adresářům */lib*, */opt* a */usr*. Poslední záznam slouží k připojení implicitního úložiště na lokálním disku. Aby došlo při startu k připojení CD mechaniky, bylo nutné upravit soubor */etc/rc.sysinit*. Pro správné provedení kroků při zavádění systému bylo nutné proházet pořadí provádění jednotlivých kroků. Celý upravený soubor */etc/rc.sysinit* se nachází v příloze B.

Příklad č. 8: Záznam přidáný do souboru /etc/rc.sysinit

```
/bin/mount /dev/cdrom /mnt/cdrom -o ro -t  
iso9660
```

Pro správné fungování souborového systému UDEV je potřeba vytvořit přípojné uzly *null* a *console* v adresáři */dev* (příloha C příklad č. 18).

Takto vytvořený obraz jsem odpojil od systému a zkomprimoval pomocí programu *gzip* (příloha C příklad č. 19).

Do adresářové struktury live CD jsem do adresáře *isolinux* překopíroval obraz kernelu *vmlinuz*. Pomocí programu *mkisofs* jsem vytvořil obraz CD ve formátu ISO (příloha C příklad č. 20).

Takto vytvořený obraz CD disku lze následně vypálit kterýmkoli vypalovacím softwarem. CD vypálené z obrazu se nachází v příloze D.

4.5 Úprava LiveCD

Vytvořené liveCD je možné pro potřeby uživatele upravit. Pro úpravu je potřeba mít instalovaný systém GNU/Linux s balíky *cdrkit* (či *cdrtools*) a *e2fsprogs*.

Z *cd* vykopírujeme celou adresářovou strukturu do pomocného adresáře. V podadresáři *isolinux* pomocí programu *gzip* rozbalíme soubor *miniroot.gz* do souboru *miniroot*. Tento soubor pak pomocí příkazu *mount* (kapitola 4.4 příklad 6) připojíme souborový systém obsažen v tomto souboru.

V tomto souborovém systému můžeme provádět potřebné změny. Pokud se změny týkají adresářů */usr* či */opt* je nutno pracovat v podadresáři *system* v kopii adresářové struktury CD.

Pro změnu jádra je potřeba překopírovat nové jádro do podadresáře *isolinux* a pojmenovat ho *kernel26* či poupravit soubor *isolinux.cfg* v tomtéž podadresáři, aby ukazoval na nové jádro systému.

Po provedení všech změn odpojíme soubor *miniroot* a opět jej pomocí programu *gzip* zkomprimujeme do souboru *miniroot.gz*. Je potřeba zkontrolovat, zda velikost souboru *miniroot.gz* nepřesáhla velikost ramdisku definovanou v souboru *isolinux.cfg*. Případně upravit velikost ramdisku aby postačovala.

Pro vytvoření iso obrazu je potřeba spustit z pomocného adresáře program *mkisofs* (příloha C, příklad 20).

5 Závěr

Zadaný úkol se podařilo během práce splnit v plném rozsahu.

V rámci práce byly zhodnoceny již existující softwarové produkty, komerční i nekomerční. Došlo k porovnání výhod a nevýhod těchto produktů, vzhledem k výše zhodnoceným nevýhodám nedošlo k nasazení těchto produktů v síťové laboratoři.

Proto byl v rámci práce také vytvořen vlastní systém přímo na míru potřebám síťové laboratoře. Tento systém je však možno použít i pro obecné zálohování diskových oddílů ať už pro komerční či soukromé použití.

Během vývoje byla přidána možnost volby zda použít komprimovaný či nekomprimovaný formát souboru. Při použití komprimovaného formátu totiž dochází ke značnému zatěžování procesoru a výsledná doba zálohování či obnovy se enormě prodlužuje.

Systém prošel několika stádii vývoje, kdy do něj byly postupně přidávány možnosti přesně podle potřeb užití v síťové laboratoři.

Díky otevřenosti použitých nástrojů je daný systém velmi snadno modifikovatelný a lze jej tedy přizpůsobit potřebám plynoucích z budoucího vývoje. K práci jsou přiloženy dvě CD. Jedno obsahuje funkční verzi systému a druhé upravené skripty, zdrojové kódy a tento dokument v elektronické podobě.

6 Seznam obrázků a tabulek

6.1 Seznam obrázků

1. Rozdělení diskových oddílů na počítači v síťové laboratoři. Zdroj: vlastní, strana 11
2. Paragon system recovery. Zdroj: www.paragon-software.com, 05-04-2007, strana 14
3. Partimage. Zdroj: www.parimage.org, 31-03-2007, strana 19
4. Struktura uživatelského rozhraní. Zdroj: vlastní, strana 21
5. Záloha na lokální úložiště. Zdroj: vlastní, strana 22
6. Záloha na síťové úložiště, NFS. Zdroj: vlastní, strana 23
7. Záloha na síťové úložiště, SMB. Zdroj: vlastní, strana 23
8. Rozdělení diskových oddílů ve virtuálním PC. Zdroj: vlastní, strana 26
9. Úvodní nabídka systému. Zdroj: vlastní, strana 31
10. Průběh zálohování diskového oddílu hda1 do obrazu WindowsXP.img.gz v úložišti. Zdroj: vlastní, strana 32

6.2 Seznam tabulek

1. Podpora souborových systémů v programu Partimage, Zdroj : www.partiamge.org, 04-04-2007, strana 18

7 Použité zkratky a pojmy

GUID Partition Table (GPT) – Jedná se o standart vzhledu tzv. „partition table“ na pevném disku. Jedná se o součást standartu Extensible Firmware Interface vyvinutém společností Intel jako náhrada zastaralého modelu PC BIOS. GPT nahrazuje master boot record používaný v rámci PC BIOSu.

LiveCD – či tzv. „živé CD“. Představuje CD disk, ze kterého je možno nastartovat a provozovat plnohodnotný operační systém.

DHCP – Jedná se o protokol umožňující klientům na počítačové síti získat od serveru parametry svého síťového připojení jako jsou IP adresa, maska podsítě, výchozí brána či adresy DNS serverů.

8 Použité zdroje

1. SOBELL, M. *Linux, praktický průvodce*. 1. vyd. Praha: Computer press, 1999. ISBN 80-7226-190-8
2. PUŽMANOVÁ, R. *TCP/IP v kostce*. 1. vyd. České Budějovice: Kopp, 2004. ISBN 80-7232-236-2
3. POLZER, J. *GNU Emacs a VIM: kapesní přehled*. 1. vyd. Praha: Computer press, 2005. ISBN 80-251-0782-5
4. KOLEKTIV *Linux dokumentační projekt*. 3. vyd. Praha: Computer press, 2003. ISBN 80-7226-761-3
5. ArchWiki [online].2007 [cit. 2007-05-08]. Dostupný online: < <http://wiki.archlinux.org/> >
6. ArchLinux [online].2007 [cit. 2007-05-08]. Dostupný online: < <http://www.archlinux.org/> >
7. Wikipedia, the free Encyklopedia [online].2007 [cit.2007-05-08]. Dostupný online: <<http://en.wikipedia.org> >
8. LQWiki [online].2007 [cit. 2007-04-03]. Dostupný online: < <http://wiki.linuxquestions.org/> >
9. THOMAS, W. *Activating numlock during boot time* [online].2007 [cit. 2007-05-08]. Dostupný online: < <http://www.userlocal.com/tips/activatingnumlock.php> >
10. SCHRODER, C. *System recovery whith Knoppix* [online].2007 [cit. 2007-09-29]. Dostupný online: < <http://www-128.ibm.com/developerworks/linux/library/l-knoppix.html> >

9 Seznam příloh

- A. soubor `sys_rec.sh`
- B. soubor `rc.sysinit`
- C. příklady použitých příkazů
- D. CD s použitelným systémem
- E. CD obsahující soubory `sys_rec.sh`, `rc.sysinit`, `autologinroot.c` a tento dokument v elektronické podobě.

Příloha A – soubor sys_rec.sh

```
##! /bin/sh

#zapnuti numlock

setleds +num

#nacteni nastaveni site pres DHCP

dhcpcd

#odemknuti CD mechaniky

echo 0 > /proc/sys/dev/cdrom/lock

#funkce pro zobrazeni menu

menu() {

    echo "Nastaveni site (eth0): "

    ifconfig eth0

    echo "Hint -- "

    echo "volbou 3 se dostanete do konzole

systemu. Login == root, heslo == rootroot"

    echo "Prikazem menu z konzole se

dostanete zpet do menu."

    echo ''

    #zobrazeni aktualne pripojeneho uloziste

    case $STORAGE in

        1)echo "STORAGE : local";;

        2)echo "STORAGE : NFS/"${SERVER};;
```

```

3)echo "STORAGE : SAMBA/"${SERVER};;

*)echo "Tak sem se to nemelo vubec
dostat!!!";;

esac

echo "1 - Obnovit systemovy oddil";
echo "2 - Zalohovat systemovy oddil";
echo "3 - Pripojit uloziste"
echo "4 - Nastaveni site"
echo "0 - Systemova konzole"
echo "z - automaticka zaloha"
echo "o - automaticka obnova"
echo "k - Konec";

}

#funkce pro obnoveni diskoveho oddilu z
uloziste
recovery() {
    echo "Soubory v ulozisti : "
    #vypsani souboru v ulozisti
    ls -lahF /storage
    echo ''
    echo
    "*****"
    echo ''
    echo "0 - navrat do menu"
    #nacteni nazvu souboru

```

```

        read -p "Zadej nazev obrazu (napr.
WinXP_Lin.img): [default.img]" SOURCE_FILE;
        if [ ${#SOURCE_FILE} -eq 0 ]
        then
                SOURCE_FILE="default.img"
        fi
        case ${SOURCE_FILE} in
                0) return;;
        esac
        #overeni existence souboru
        if [ -f /storage/${SOURCE_FILE} ]
        then
                echo ""
                echo "Detekovane diskove oddily : "
                ls /dev/hd*
                echo ""
                #nacteni ciloveho diskoveho oddilu
                read -p "Zadej cilovou
partition(napr. hda1) : [hda]" DEST_FILE
                case ${DEST_FILE} in
                        0) return;;
                esac
                if [ ${#DEST_FILE} -eq 0 ]
                then
                        DEST_FILE="hda"
                fi
                #overeni existence diskoveho oddilu

```

```

if [ -e /dev/${DEST_FILE} ]
then
    read -p "Je cilovy obraz
komprimovan?(Y,n) " $COMPRESS
    case ${COMPRESS} in
        n,N)
            echo "Obnovuji systemovy
oddil ${DEST_FILE} ze souboru
/storage/${SOURCE_FILE} ..."
            echo "Mejte strpeni ..."
            #obnova dat
            dcfldd
if=/storage/${SOURCE_FILE}
of=/dev/${DEST_FILE} bs=8192 sizeprobe=of
            echo "System obnoven."
            read -p "Pro pokracovani
stiskni ENTER." POM
            ;;
        *)
            echo "Obnovuji systemovy
oddil ${DEST_FILE} ze souboru
/storage/${SOURCE_FILE} ...";
            echo "Mejte strpeni ...";
            #dekomprimovani a
nasledne prekopirovani dat do diskoveho oddilu
            gzip -dc
/storage/${SOURCE_FILE} | dcfldd
of=/dev/${DEST_FILE} bs=8192 sizeprobe=of;
            echo "System obnoven."

```

```

                read -p "Pro pokračovani
stiskni ENTER" POM

                ;;

            esac

        else

            echo "Cilova partition
neexistuje!!!"

            fi

    else

        echo "Soubor s obrazem nenalezen!!!"

        fi

}

#funkce pro zalohovani diskoveho oddilu
backup() {

    echo "Detekovane diskove oddily : "

    #vypis diskovych oddilu
    ls -lahF /dev/hd*

    echo "0 - navrat do menu"

    #nacteni nazvu diskoveho oddilu

    read -p "Zadej partition k
zalohovani(napr. hda1) : [hda]" SOURCE_FILE;

    case ${SOURCE_FILE} in

        0) clear;return;;

    esac

    if [ ${#SOURCE_FILE} -eq 0 ]

```



```

then
    SOURCE_FILE="hda"
fi

#overeni existence diskoveho oddilu
if [ -e /dev/${SOURCE_FILE} ]
then
    #nacteni nazvu ciloveho obrazu
    read -p "Zadej nazev ciloveho obrazu
(napr. WinXP_Lin.img): [default.img]"
DEST_FILE

    case ${DEST_FILE} in
        0)clear;return;;
    esac

    if [ ${#DEST_FILE} -eq 0 ]
    then
        DEST_FILE="default.img"
    fi

    read -p "Komprimovat cilovy
obraz?(Y,n)" $COMPRESS

    case ${COMPRESS} in
        n,N)

            echo "Zalohuji systemovy oddil
/dev/${SOURCE_FILE} do souboru
/storage/${DEST_FILE} ... "

            echo "Mejte strpeni ... "

            if dcfldd if=/dev/${SOURCE_FILE}
of=/storage/${DEST_FILE} bs=8192 sizeprobe=of
then

```

```

        echo "System uspesne
zaloховan."
    else
        "Nepodarilo se zaloховat
system!!!"
    fi
    read -p "Pro pokračovani stiskni
ENTER" POM

    ;;
    *)

        echo "Zalohuji systemovy oddil
/dev/${SOURCE_FILE} do souboru
/storage/${DEST_FILE} ...";

        echo "Mejte chvili strpeni
...";

        #prekopirovani a zkomprimovani dat

        if dcfldd
if=/dev/${SOURCE_FILE} bs=8192 sizeprobe=if |
gzip > /storage/${DEST_FILE}

        then

            echo "System uspesne
zaloховan,"

        else

            echo "System se nepodarilo
zaloховat!!!"

        fi

        read -p "Pro pokračovani
stiskni ENTER" POM

        ;;

```

```

        esac

    else

        echo "Zdrojova partition
neexistuje!!!"

        fi

    }

#funkce pro pripojeni uloziste, parametr udava
jake uloziste se ma pripojit, 1 - lokalni
diskovy oddil, 2 - NFS. 3 - SMB
mount_storage() {

    umount /storage

    case $1 in

        #lokalni diskovy oddil
        1)clear;

            echo "Diskove oddily : "

            #vypsani diskovych oddilu
            ls /dev/hd*

            echo "Zadej diskovy oddil k
pripojeni. Tento oddil nebude mozne
zalohovat.";echo "0 -- navrat do menu.";

            #nacteni nazvu diskoveho oddilu

            read -p "Volba (napr. hda1):
[hda5]" DEVICE

            case $DEVICE in

                0)return;;

            esac

            if [ ${#DEVICE} -eq 0 ]

```

```

then

    DEVICE="hda5"

fi

#pripojeni diskoveho oddilu

mount /dev/${DEVICE} /storage
2>/dev/null && echo "Uloziste pripojeno." ||
echo "Nepodarilo se pripojit uloziste!!!";

#NFS

#nacteni informaci o serveru

2)read -p "Zadej adresu serveru
(napr. 192.168.1.1): [192.168.1.1]" SERVER;

if [ ${#SERVER} -eq 0 ]

then

    SERVER=192.168.1.1

fi

echo "Adresare sdilene na serveru
: "

showmount -e ${SERVER}

echo ""

read -p "Zadej adresar na serveru
:" SERVER_DIR

echo "Pipojuji NFS, mejte chvili
strpeni ..."

#pripojeni souboroveho systemu

mount -t nfs
${SERVER}:${SERVER_DIR} /storage 2>/dev/null
&& echo "Uloziste pripojeno." || echo
"Nepodarilo se pripojit uloziste!!!";

```

```

#SMB

#nacteni adresy serveru, pokud DHCP
prideli i DNS server, lze zadat DNS jmeno

3)read -p "Zadej adresu serveru
(napr. 192.168.1.1): [192.168.1.1]" SERVER

if [ ${#SERVER} -eq 0 ]
then

SERVER=192.168.1.1

fi

echo
"*****"

echo " Slozky sdilene na
serveru "

echo
"*****"

echo ""

#vypis sdilenych slozek na serveru
smbclient -N -L ${SERVER}

echo ""

echo
"*****"

#nacteni sdilene slozky
read -p "Zadej nazev sdilene
slozky : " SERVER_DIR

#pripojeni souboroveho systemu
mount -t smbfs
//${SERVER}/${SERVER_DIR} /storage && echo
"Uloziste pripojeno" || echo "Nepodarilo se
pripojit uloziste!!!";

```

```

        *)echo "Chybne volani funkce
mount_storage()!!!";;

        esac
    }

#funkce pro nastaveni site
set_network(){
    while true;
    do
        echo "Detakovana sitova rozhrani : "
        #vypis detekovanych sitovych karet
        ls /dev/eth*
        echo "Zvol nastaveni site : "
        echo "0 - navrat do menu"
        echo "1 - staticke"
        echo "2 - DHCP"
        #nacteni volby
        read -p "Volba : " NET_SET
        case ${NET_SET} in
            #navrat do menu
            0) clear;return;;
            #staticke nastaveni
            1) read -p "Zadej IP adresu
(napr. 192.168.1.254): " IP_ADDR

```

```

        read -p "Zadej masku (napr.
255.255.255.0): " IP_MASK
        if ifconfig eth0 ${IP_ADDR}
netmask ${IP_MASK} 2>/dev/null
        then
            echo "Adresa nastavena
uspesne.";
        else
            echo "Nepodarilo se
nastavit sit!!!";
        fi
        read -p "Zadej adresu DNS
serveru (napr. 192.168.1.1):" DNS_ADDR
        echo "nameserver
"${DNS_ADDR} >> /etc/resolv.conf;
        read -p "Zadej adresu
vychozi brany (napr. 192.168.1.1): " GW_ADDR
        if route add default gw
${GW_ADDR} 2>/dev/null
        then
            clear;echo "Brana
Uspesne nastavena."; return;
        else
            clear;echo "Nepodarilo
se nastavit branu!!!";return;
        fi;;
#nastaveni pomoci DHCP
2)if dhcpcd 2>/dev/null
    then

```

```
                clear;echo "Adresa
nastavena pomoci DHCP.";
                else
                        clear;echo "Nepodarilo se
ziskat adresu daneho zarizeni!!!";
                fi
                break;;
                *)clear;echo "Neplatna
volba!!!";;
                esac
        done
}
```

```
#STORAGE - Urceni uloziste, 1 ... local , 2
... NFS, 3 ... Samba
#default - 1
STORAGE=1;
```

```
clear
```

```
while true;
do
        #zobrazeni menu
        menu
        #nacteni volby
```



```

read -p 'Volba : ' OPTION;
#vyhodnoceni volby
case $OPTION in
    #obnova
    1)  clear; recovery;;
    #zaloha
    2)  clear; backup;;
    #vstup do konzole
    0)  clear;
        echo ""
            echo "Pro navrat do nabidky
pouzijte prikaz menu.";
            echo "Pro praci s diskovymi
oddily pouzijte prikaz cfdisk"
            echo "Prikaz mc spusti souborovy
manazer"
            echo "Pro sledovani site
pouzijte prikaz iftop"
            echo ""
            break
        ;;
    #pripojeni uloziste
    3)  clear;
        while true;
        do
            echo "Zvol uloziste:"

```

```

        echo "1 - local"; echo "2 -
NFS"; echo "3 - Samba"; echo "0 - navrat do
menu"

        read -p "Uloziste : "
STORAGE_TMP;

        case ${STORAGE_TMP} in

                0)clear;break;;

                1|2|3) STORAGE=${STORAGE_TMP};mount_storag
e ${STORAGE};break;;

                *)clear; echo
"Neplatna volba!!!";;

        esac

        done;;

        #nastaveni site

        4)  clear;set_network;;

        #automaticka zaloha

        z)  clear

                #pokus o pripojeni diskety

                if mount /dev/fd0 2>/dev/null

                then

                        #uspesne pripojeni, nacteni
hodnot z diskety

                                source

                                /mnt/floppy/autoconf.sh

                                else

                                        #neuspesne pripojeni,
nacteni implicitnich hodnot

```

```

        source
/mnt/cdrom/system/autoconf.sh

        fi

        umount /storage

        #pripojeni uloziste podle
parametru

        echo "Pripojuji uloziste. Mejte
strpeni ..."

        mount -t nfs
${AUTO_SERVER}:${AUTO_STORAGE} /storage

        echo "Provadim zalohu
${AUTO_DISK} do obrazu ${AUTO_IMAGE}"

        echo "Mejte chvili strpeni ..."

        #zaloha diskoveho oddilu

        dcfldd if=${AUTO_DISK}
of=/storage/obrazyHD/${AUTO_IMAGE} bs=8192
sizeprobe="if"

        umount /mnt/floppy 2>/dev/null

        ;;

#automaticka obnova
o) clear

        #pokus o pripojeni diskety

        if mount /dev/fd0 2> /dev/null

        then

                #uspesne pripojeni, nacteni
hodnot z diskety

                source
/mnt/floppy/autoconf.sh

        else

```

```

                                #neuspesne pripojeni,
nacteni implicitnich hodnot

                                source

/mnt/cdrom/system/autoconf.sh

                                fi

                                umount /storage

                                #pripojeni uloziste podle
parametru

                                echo "Pripojuji uloziste. Mejte
strpeni ..."

                                mount -t  nfs
${AUTO_SERVER}:${AUTO_STORAGE}  /storage

                                echo "Provadim obnovu
${DISK_AUTO} z obrazu ${AUTO_IMAGE}"

                                echo "Mejte chvili strpeni ..."

                                #obnova diskoveho oddilu

                                dcfldd

if=/storage/obrazyHD/${AUTO_IMAGE}
of=${AUTO_DISK} bs=8192 sizeprobe=of

                                umount /mnt/floppy 2>/dev/null

                                ;;

                                #konec

                                k)  umount /mnt/cdrom;eject;
shutdown -r now; break;;

                                *)  clear; echo "Neplatna volba";;

esac

done

```

Příloha B - soubor rc.sysinit

```
#!/bin/sh

#

# /etc/rc.sysinit

#

. /etc/rc.conf

. /etc/rc.d/functions

echo " "

printhl "System recovery based on Arch Linux
v0.7.1 ${C_OTHER}(${C_H2})Noodle${C_OTHER}\n"
printhl "${C_H2}http://www.archlinux.org"
printhl "Copyright 2002-2006 Judd Vinet"
printhl "Distributed under the GNU General
Public License (GPL)"

printsep

# start up our mini logger until syslog takes
over

/sbin/minilogd

# anything more serious than KERN_WARNING goes
to the console

/bin/dmesg -n 3

# mount /proc and /sys
```

```

mount -n -t proc none /proc

[ "`grep sysfs /proc/filesystems`" ] && mount
-n -t sysfs none /sys

if [ -e /dev/.devfsd -a -x /sbin/devfsd ];
then

    # Looks like devfs is running, use it
    status "Starting DevFS Daemon"
/sbin/devfsd /dev

elif [ -x /etc/start_udev -a -d /sys/block ];
then

    # We have a start_udev script and /sys
appears to be mounted, use UDev
    status "Starting UDev Daemon"
/etc/start_udev

else

    # Static /dev, our last resort
    status "Using static /dev filesystem"
/bin/true

fi

# If using an encrypted root fs, we should
find the root dev in the initrd

if [ -e /initrd/dev/mapper/root ]; then

    mkdir /dev/mapper 2>/dev/null
    cp -a /initrd/dev/mapper/root
/dev/mapper/root

fi

```

```

# If necessary, find md devices and manually
assemble RAID arrays

if [ -f /etc/mdadm.conf -a "`grep ^ARRAY
/etc/mdadm.conf 2>/dev/null`" ]; then

    if [ -d /initrd/dev ]; then

        # udev won't create these md nodes,
so we steal them from the initrd

        for i in `grep ^ARRAY
/etc/mdadm.conf | awk '{print $2}'`; do

            cp -a /initrd/$i /dev/

        done

    fi

    status "Activating RAID arrays"
/sbin/mdadm --assemble --scan

fi

# Unmount and free up the old initrd if it
exists

if [ -f /initrd/linuxrc ]; then

    stat_busy "Freeing memory from Initial
Ramdisk"

    umount /initrd 2>/dev/null

    /sbin/blockdev --flushbufs /dev/rd/0
2>/dev/null

    stat_done

fi

# Set up non-root encrypted partition mappings

```

```

if [ -f /etc/crypttab ]; then
    CS=/sbin/cryptsetup.static
    while read line; do
        [ "$line" = "" ] && continue
        [ "`echo $line | grep ^#`" ] &&
continue
        # remove redundant whitespace and
split up tokens
        line=`echo $line | awk '{print $0}'`
        cname=`echo $line | cut -d' ' -f1`
        csrc=`echo $line | cut -d' ' -f2`
        cpass=`echo $line | cut -d' ' -f3-`
        [ "$cpass" = "" ] && continue

        action=create
        $CS isLuks $csrc 2>/dev/null &&
action=luksOpen

        if [ "`echo $cpass | grep \\\"`" ];
then
            # Trim off the quotes around the
password string
            cpass2=`echo $cpass | sed -e
's|^"||' -e 's|"$||'`
            # For some fun reason, the
parameter ordering varies for
            # LUKS and non-LUKS devices.
Joy.

```



```

        if ` $CS isLuks $csrc
2>/dev/null`; then
            echo "$cpass2" | $CS
luksOpen $csrc $cname >/dev/null
        else
            echo "$cpass2" | $CS create
$cname $csrc >/dev/null
        fi
    else
        if ` $CS isLuks $csrc
2>/dev/null`; then
            $CS -d $cpass luksOpen
$csrc $cname >/dev/null
        else
            $CS -d $cpass create $cname
$csrc >/dev/null
        fi
    fi
done </etc/crypttab
fi

if [ "$USELVM" = "yes" -o "$USELVM" = "YES" ];
then
    if [ -f /etc/lvmtab -a -x /sbin/vgchange
]; then
        # Kernel 2.4.x, LVM1 groups
stat_busy "Activating LVM1 groups"
/sbin/vgchange -a y

```

```

        stat_done
    elif [ -x /sbin/lvm -a -d /sys/block ];
then
    # Kernel 2.6.x, LVM2 groups
    stat_busy "Activating LVM2 groups"
    /sbin/lvm vgscan --
ignorelockingfailure
    /sbin/lvm vgchange --
ignorelockingfailure -a y
    stat_done
fi
fi

status "Activating Swap" /sbin/swapon -a

status "Mounting Root Read-only" /bin/mount -n
-o remount,ro /

FORCEFSCK=

[ -f /forcefsck ] && FORCEFSCK="-- -f"

if [ -x /sbin/fsck ]; then
    stat_busy "Checking Filesystems"
    if [ "`cat /proc/cmdline | grep quiet`"
]; then
        /sbin/fsck -A -T -C -a $FORCEFSCK
>/dev/null 2>&1
    else

```

```

        /sbin/fsck -A -T -C -a $FORCEFSCK

fi

if [ $? -gt 1 ]; then

    stat_fail

    echo

    echo "***** FILESYSTEM
CHECK FAILED *****"

    echo "*"

*"

    echo "* Please repair manually and
reboot. Note that the root *"

    echo "* file system is currently
mounted read-only. To remount *"

    echo "* it read-write type: mount -
n -o remount,rw /          *"

    echo "* When you exit the
maintenance shell the system will   *"

    echo "* reboot automatically.
*"

    echo "*"

*"

    echo

"*****"
*****"

    echo

    /sbin/sulogin -p

    echo "Automatic reboot in
progress..."

    /bin/umount -a

```

```

        /bin/mount -n -o remount,ro /
        /sbin/reboot -f
        exit 0
    fi

    stat_done
fi

status "Updating Module Dependencies"
/sbin/depmod -A

if [ -f /var/run/random-seed ]; then
    stat_busy "Initializing Random Seed"
    /bin/cat /var/run/random-seed
>/dev/urandom
    stat_done
fi

# make sure / gets written to /etc/mtab
#prenacteni / aby mohl bezet hwdetect ...
/bin/mount -o remount,rw /

#nacteni modulu, pred mountem to zajisti ze
budou videt ide zarizeni
if ! [ "$load_modules" = "off" ]; then
    if [ -f /proc/modules ]; then
        stat_busy "Loading Modules"
        for mod in "${MODULES[@]}"; do
            if [ "$mod" = "${mod#!}" ]; then

```

```

        /sbin/modprobe $mod
    fi
done
    if [ "$MOD_AUTOLOAD" = "yes" -o
"$MOD_AUTOLOAD" = "YES" ]; then
        stat_busy "Loading hwdetect"
        export
BLACKLIST="${MOD_BLACKLIST[*]}"
        /sbin/hwdetect --load-modules
    fi
    stat_done
    stat_done
fi

fi

stat_busy "Mounting Local Filesystems"
/bin/mount /dev/cdrom /mnt/cdrom -o ro -t
iso9660
/bin/mount -n -o remount,rw /
/bin/rm -f /etc/mtab*
# make sure / gets written to /etc/mtab
/bin/mount -o remount,rw /
# re-mount /proc and /sys so they can be
written to /etc/mtab
umount /proc && mount -t proc none /proc

```

```

[ "`grep sysfs /proc/filesystems`" ] && umount
/sys && mount -t sysfs none /sys

# now mount all the local filesystems

/bin/mount -a -t
nonfs,nosmbfs,nocifs,noncpfs,nosysfs,nousbfs

stat_done

stat_busy "Configuring System Clock"
if [ "$HARDWARECLOCK" = "UTC" ]; then
    /sbin/hwclock --utc --hctosys
else
    /sbin/hwclock --localtime --hctosys
fi

if [ ! -f /var/lib/hwclock/adjtime ]; then
    echo "0.0 0 0.0" >
/var/lib/hwclock/adjtime
fi

if [ "$TIMEZONE" != "" ]; then
    /bin/ln -sf /usr/share/zoneinfo/$TIMEZONE
/etc/localtime
fi

stat_done

stat_busy "Removing Leftover Files"
/bin/rm -f /etc/nologin &>/dev/null

/bin/rm -f /etc/shutdownpid &>/dev/null

/bin/rm -f /var/lock/* &>/dev/null

```

```

/bin/rm -rf /tmp/* /tmp/.* &>/dev/null
/bin/rm -f /forcefsck &>/dev/null
(cd /var/run && find . ! -type d -exec rm -f -
- {} \; )
: > /var/run/utmp
# Keep {x,k,g}dm happy with xorg
mkdir /tmp/.ICE-unix && chmod 1777 /tmp/.ICE-
unix
mkdir /tmp/.X11-unix && chmod 1777 /tmp/.X11-
unix
stat_done

#status "Updating Shared Library Links"
/sbin/ldconfig

if [ "$HOSTNAME" != "" ]; then
    status "Setting Hostname: $HOSTNAME"
    /bin/hostname $HOSTNAME
fi

# Set the NIS domain name, if necessary
[ -f /etc/conf.d/nisdomainname ] && .
/etc/conf.d/nisdomainname

if [ "$NISDOMAINNAME" != "" ]; then
    status "Setting NIS Domain Name:
$NISDOMAINNAME" /bin/nisdomainname
$NISDOMAINNAME
fi

```

```

status "Updating Module Dependencies"
/sbin/depmod -A

if [ -f /var/run/random-seed ]; then
    stat_busy "Initializing Random Seed"
    /bin/cat /var/run/random-seed
>/dev/urandom
    stat_done
fi

if [ "$KEYMAP" != "" ]; then
    status "Loading Keyboard Map: $KEYMAP"
    /bin/loadkeys -q $KEYMAP
fi

# Set user defined locale
if [ "$LOCALE" != "" ]; then
    stat_busy "Setting Locale: $LOCALE"
    echo "export LANG=$LOCALE"
>/etc/profile.d/locale.sh
    /bin/chmod 755 /etc/profile.d/locale.sh
    stat_done

# If locale is *.utf set console to Unicode
mode

```



```

        if [ "`echo $LOCALE | /bin/grep -i utf`"
]; then

        stat_busy "Setting Consoles to UTF-
8"

        /usr/bin/kbd_mode -u

        /usr/bin/dumpkeys | /bin/loadkeys --
unicode

        echo 'echo -ne "\e%G"'
>>/etc/profile.d/locale.sh

        stat_done

    fi

else

    rm -f /etc/profile.d/locale.sh

fi

if [ "$CONSOLEFONT" != "" ]; then

    stat_busy "Loading Console Font:
$CONSOLEFONT"

    for i in `seq 1 12`; do

        if [ "$CONSOLEMAP" != "" ]; then

            /usr/bin/setfont -m $CONSOLEMAP
$CONSOLEFONT -C /dev/vc/${i}

        else

            /usr/bin/setfont $CONSOLEFONT -C
/dev/vc/${i}

        fi

        echo -ne "\e(K" >/dev/vc/${i}

    done

```

```

        stat_done

fi

# Load USB support

/sbin/modprobe usbcore >/dev/null 2>&1

[ "`grep usbfs /proc/filesystems`" ] && mount
-t usbfs none /proc/bus/usb

stat_busy "Mounting Local Filesystems"

/bin/mount /dev/cdrom /mnt/cdrom -o ro -t
iso9660

/bin/mount -n -o remount,rw /

/bin/rm -f /etc/mtab*

# make sure / gets written to /etc/mtab

/bin/mount -o remount,rw /

# re-mount /proc and /sys so they can be
written to /etc/mtab

umount /proc && mount -t proc none /proc

[ "`grep sysfs /proc/filesystems`" ] && umount
/sys && mount -t sysfs none /sys

# now mount all the local filesystems

/bin/mount -a -t
nonfs,nosmbfs,nocifs,noncpfs,nosysfs,nousbfs

stat_done

# Screen blanks after 15 minutes idle time

#/usr/bin/settem -blank 15

```

```
# Load modules from the MODULES array defined
in rc.conf

#spusteni hwdetect

if ! [ "$load_modules" = "off" ]; then
    if [ "$MOD_AUTOLOAD" = "yes" -o
"$MOD_AUTOLOAD" = "YES" ]; then
        stat_busy "Loading hwdetect"
        export
BLACKLIST="${MOD_BLACKLIST[*]}"
        /sbin/hwdetect --load-modules
    fi
    stat_done
fi

# End of file

# vim: set ts=2 noet:
```

Příloha C – příklady použitých příkazů

Příklad č. 1: Změny provedené v souboru /etc/rc.conf

```
HOSTNAME="Systém recovery"  
eth0="dhcp"  
gateway="dhcp"
```

Příklad č. 2: Záznam v /etc/fstab pro připojení /storage

```
/dev/hda5 /storage ext3 defaults 0 0
```

Příklad č. 3: Doinstalování balíčků smbclient a dcfldd

```
#pacman -S --refresh  
#pacman -S smbclient  
#pacman -S nfs-utils  
#pacman -S dcfldd
```

Příklad č. 4: Doinstalování balíčků mc a iftop

```
#pacman -S mc  
#pacman -S iftop  
#pacman -S eject  
#pacman -S openssh
```

Příklad č. 5: Příkaz přidáný na konec souboru /root/.bashrc

```
/root/sys_rec.sh
```

Příklad č. 6: Program v jazyce C vykonávající automatické přihlášení uživatele root po startu systému.

```
#include <unistd.h>

int main(){

    execlp("login","login","-f","root",0);

    return 0;

}
```

Příklad č. 7: Upravený řádek v souboru /etc/inittab.

```
c1:2345:respawn:/sbin/agetty -n -l
/usr/local/sbin/autologinroot 38400 vc/1 linux
```

Příklad č. 8: Nastavení SUID bitu souboru sys_rec.sh pomocí příkazu chmod.

```
#chmod u+s ~/sys_rec.sh
```

Příklad č. 9: Aliasy přidané do souboru /root/.bashrc

```
alias menu='~/root/sys_rec.sh'

alias l='ls -laF'
```

Příklad č. 10: Nastavení IP adresy 192.168.1.222 s maskou podsítě 255.255.255.0 rozhraní eth0

```
ifconfig eth0 192.168.1.222 netmask
255.255.255.0
```

Příklad č. 11: Nastavení routeru s IP adresou 192.168.1.1 jako výchozí brány.

```
route add default gw 192.168.1.1
```

Příklad č. 12: Restart systému pomocí příkazu shutdown

```
shutdown -r now
```

Příklad č. 13: Instalace balíčků cdrkit a e2fsprogs

```
#pacman -S cdrkit  
#pacman -S e2fsprogs
```

Příklad č. 14: Konfigurační soubor zavaděče isolinux.cfg

```
prompt 0  
timeout 0  
display boot.msg  
default vmlinuz initrd=/miniroot.gz  
ramdisk_size=100000 load_ramdisk=1  
prompt_ramdisk=0 root=/dev/ram0
```

Příklad č. 15: Vytvoření obrazu

```
#touch miniroot  
#mkfs.ext2 miniroot 100000
```

Příklad č. 16: Vytvoření symbolických odkazů na adresáře připojené v /mnt/cdrom/system.

```
#ln -sf /mnt/cdrom/opt opt  
#ln -sf /mnt/cdrom/usr usr  
#ln -sf /mnt/cdrom/lib lib
```

Příklad č. 17: Záznamy v souboru /etc/fstab

```
/dev/root / ext2 defaults 0 0  
none /proc proc defaults 0 0
```

```
/dev/cdrom /mnt/cdrom iso9660 ro, user,  
noauto, unhide 0 0  
  
/dev/hda5 /storage auto defaults 0 0
```

Příklad č. 18: Vytvoření přípojných uzlů v adresáři /dev

```
#mknod -m 660 console c 5 1  
  
#mknod -m 660 null c 1 3
```

Příklad č. 19: Odpojení obrazu a zkomprimování pomocí příkazu gzip

```
#umount /mnt/image  
  
#gzip miniroot miniroot.gz
```

Příklad č. 20: Vytvoření obrazu CD

```
mkisofs -o /storage/System_recovery.iso -R -V  
"System recovery 1.0" -T -b  
isolinux/isolinux.bin -c isolinux/boot.cat -  
no-emul-boot -boot-load-size 4 -boot-info-  
table -A "System recovery 1.0" .
```

ÚDAJE PRO KNIHOVNICKOU DATABÁZI

Název práce	Systém pro automatické obnovování systémů na PC v síťové laboratoři
Autor práce	Ladislav Šorčík
Obor	Informační technologie
Rok obhajoby	2007
Vedoucí práce	Ing. Martin Dobrovolný
Anotace	Cílem práce je vytvořit systém umožňující snadnou obnovu systémových diskových oddílů na počítačích v síťové laboratoři. A to jak systémy na UNIXovém základě, tak systémy Windows. Systém bude umožňovat tvorbu obrazů disků a následnou obnovu systému z těchto záloh.
Klíčová slova	záloha obnova Linux operační systém archlinux systémový oddíl