

# Konfigurace sítě s napojením na externí úložiště

Configuration of Network Attached Storage

Ivo Špičák

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2012/2013

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ivo ŠPIČÁK**  
Osobní číslo: **A10104**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Bezpečnostní technologie, systémy a management**  
Forma studia: **kombinovaná**

Téma práce: **Konfigurace Network Attached Storage**

Zásady pro vypracování:

1. Zpracujet literární rešerši na téma síťové disky.
2. Provedte instalaci a konfiguraci NAS v Linuxovém (Unix) prostředí.
3. Nakonfigurujte protokoly NFS a SMB.
4. Implementujte podporu iSCSI.
5. Navrhněte zabezpečení.
6. Nastavte Power Management.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. BIGELOW, Stephen J. **Mistrovství v počítačových sítích: správa, konfigurace, diagnostika a řešení problémů.** Vyd. 1. Překlad Petr Matějů. Brno: Computer Press, 2004, 990 s. ISBN 80-251-0178-9.
2. DOSTÁLEK, Libor. **Velký průvodce protokoly TCP/IP: Bezpečnost.** 1.vyd. Praha: Computer Press, 2001, 565 s. ISBN 80-722-6513-X.
3. ECKSTEIN, Robert. **Samba: Linux – souborový a tiskový server pro heterogenní síť.** 1. vyd. Praha: Computer Press, 2001, 375 s. ISBN 80-722-6463-X.
4. MATÝSEK, M., ADÁMEK, M., NEUMANN, P. **The security of Linux. PROCEEDINGS the 7th International Scientific – Technical Conference ? Process Control'06, 2006.**
5. SHAH, Steve. **Administrace systému Linux: překlad čtvrtého vydání.** 1. vyd. Praha: Grada, 2007, 426 s. ISBN 978-80-247-1694-7

Vedoucí bakalářské práce:

**doc. Ing. Martin Sysel, Ph.D.**

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

**25. února 2013**

Termín odevzdání bakalářské práce:

**30. května 2013**

Ve Zlíně dne 25. února 2013



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Mgr. Milan Adámek, Ph.D.  
*ředitel ústavu*

## ABSTRAKT

Bakalářská práce se věnuje konfiguraci síťových protokolů. Cílem práce je provést instalaci a následně konfiguraci *Linuxového* serveru, který slouží pro sdílení dat klientům.

Teoretická část se zabývá tematikou síťových protokolů, síťových disků. Dále pojednává o datovém úložišti na síti a vytváření velkých souborových systémů.

Praktická část zahrnuje instalaci a konfiguraci *NAS* v *Linuxovém (Unix)* prostředí, dále konfiguraci protokolů *NFS* a *SMB*, implementaci podpory *iSCSI* a návrh zabezpečení. Závěrečná část řeší nevhodnější alternativu power managementu.

Klíčová slova: *Linuxový* server, síťové protokoly, síťové disky, datové úložiště, *NAS*, *SMB*, *iSCSI*, zabezpečení disku, power management

## ABSTRACT

This Bachelor Thesis focuses on the configuration of network protocols. The target of the thesis is to execute the installation and configuration of *Linux* server that is used for clients' data sharing.

The theoretical part deals with the theme of network protocols and network disks. Furthermore it concerns data network attached storage and creation of great file systems.

The practical part includes the *NAS* installation and configuration in *Linux* environment, then *NFS* and *SMB* configuration, implementation of *iSCSI* support and the proposal of security. The final part solves the problem of power management.

Keywords: *Linux* server, network protocols, network disks, file storage, *NAS*, *SMB*, *iSCSI*, disk security, power management

Rád bych poděkoval Ing. Přemyslu Krčmářovi za konzultace vedoucí k realizaci praktické části této práce a vedoucímu bakalářské práce doc. Ing. Martinovi Syslovi, PhD. za vedení mé práce a zprostředkování zapůjčení serveru UTB.

**Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 LITERÁRNÍ ŘEŠERŠE NA TÉMA SÍŤOVÉ DISKY</b> .....	<b>11</b>
1.1 <i>NAS</i> .....	11
1.1.1 Historie.....	11
1.1.2 Principy <i>NAS</i> .....	12
1.1.3 <i>NAS</i> versus <i>SAN</i> .....	13
1.2 VÝHODA POUŽITÍ <i>RAID</i> .....	14
1.2.1 Druhy <i>RAIDu</i> .....	14
1.2.2 Paritní <i>RAID (RAID 5, RAID 6)</i> .....	17
1.3 <i>LVM</i> .....	19
1.3.1 Fyzický svazek.....	19
1.3.2 Skupina svazků.....	19
1.3.3 Logická jednotka.....	20
1.4 SÍŤOVÉ PROTOKOLY.....	21
1.4.1 Vrstvy <i>TCP/IP</i> modelu a příklady síťových protokolů.....	22
1.4.2 Síťový protokol <i>SMB – (SAMBA /CIFS)</i> .....	25
1.4.3 Síťový protokol <i>NFS</i> .....	26
1.4.3.1 Konfigurace <i>NFS</i> serveru.....	27
1.4.3.2 Konfigurace klienta.....	27
1.4.4 Protokol <i>iSCSI</i> .....	28
1.4.4.1 Bezpečnost <i>iSCSI</i> .....	29
<b>II PRAKTICKÁ ČÁST</b> .....	<b>30</b>
<b>2 INSTALACE A KONFIGURACE <i>NAS</i> V <i>LINUXOVÉM (UNIX)</i> PROSTŘEDÍ</b> .....	<b>31</b>
2.1 HW KONFIGURACE POUŽITÉHO SERVERU.....	31
2.2 POUŽITÝ OPERAČNÍ SYSTÉM.....	31
2.3 POSTUP INSTALACE.....	32
2.3.1 <i>BOOT</i> .....	32
2.3.2 Volba jazyka instalace.....	33
2.3.3 Výběr místa.....	33
2.3.4 Rozdělení disků.....	33
2.3.5 Dokončení instalace.....	35
2.3.6 Kontrola instalace.....	36
<b>3 KONFIGURACE PROTOKOLŮ <i>NFS</i> A <i>SMB</i></b> .....	<b>38</b>
3.1 PROTOKOL <i>NFS</i> .....	38
3.1.1 Server a <i>NFS</i> .....	38
3.1.2 Konfigurace <i>NFS</i> .....	39
3.1.2.1 Konfigurace souboru <i>/etc/exports</i> .....	39
3.1.2.2 Načtení souboru <i>/etc/exports</i> .....	40
3.1.3 Konfigurace klienta <i>NFS</i> .....	41
3.1.3.1 Příkaz <i>mount</i> .....	41

3.2	PROTOKOL <i>SMB</i> .....	42
3.2.1	<i>SAMBA</i> server .....	42
3.2.2	Konfigurace <i>SAMBA</i> serveru .....	43
3.2.2.1	Konfigurace části [global] .....	44
3.2.2.2	Konfigurace části [homes] .....	45
3.2.2.3	Konfigurace části [sdílený adresář] .....	46
3.2.3	Konfigurace uživatelů <i>SAMBA</i> serveru .....	48
<b>4</b>	<b>IMPLEMENTACE PODPORY <i>ISCSI</i></b> .....	<b>50</b>
4.1	INSTALACE <i>ISCSI</i> A PODPŮRNÝCH MODULŮ .....	50
4.2	KONFIGURACE /ETC/IET/IETD.CONF .....	51
4.3	PŘIPOJENÍ <i>ISCSI</i> VE <i>WINDOWS</i> .....	52
4.4	PŘIPOJENÍ <i>ISCSI</i> V <i>LINUX DEBIAN</i> .....	54
4.4.1	Kontrola připojení k serveru .....	54
4.4.2	Připojení cíle <i>iSCSI</i> klientovi .....	55
<b>5</b>	<b>NÁVRH ZABEZPEČENÍ</b> .....	<b>56</b>
5.1	BEZPEČNOST <i>SSH</i> SERVERU.....	56
5.1.1	Návrh zabezpečení <i>SSH</i> .....	56
5.2	IMPLEMENTACE SLUŽBY <i>FAIL2BAN</i> .....	58
5.2.1	Instalace a konfigurace <i>Fail2ban</i> .....	58
<b>6</b>	<b>POWER MANAGEMENT</b> .....	<b>61</b>
6.1	USPÁVÁNÍ DISKŮ .....	61
6.2	ZMĚNA TAKTOVACÍ FREKVENCE PROCESORU .....	64
6.2.1	Konfigurace .....	64
6.3	REGULACE OTÁČEK VĚTRÁKU .....	66
6.3.1	Instalace <i>lm-sensors</i> .....	66
6.3.2	Konfigurace rychlosti otáček chlazení .....	67
	<b>ZÁVĚR</b> .....	<b>69</b>
	<b>ZÁVĚR V ANGLIČTINĚ</b> .....	<b>72</b>
	<b>SEZNAM POUŽITÉ LITERATURY</b> .....	<b>75</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK</b> .....	<b>79</b>
	<b>SEZNAM OBRÁZKŮ</b> .....	<b>81</b>

## ÚVOD

21. století může být považováno za éru informací a informačních technologií. Základem všech informací jsou data a jejich sdílení je stále více aktuálním tématem. Důležité je poskytování dat v aktuálním čase, ale pro samotné poskytovatele a uživatele je prioritní především jejich ochrana neboli zabezpečení.

Téma „Konfigurace sítě s napojením na externí úložiště“, v originálním anglickém zadání *Configuration of Network Attached Storage*, je v dnešní době zcela jistě důležité a vyhledávané téma mezi IT (*Informační Technologie*) odborníky.

Bakalářská práce se věnuje konfiguraci síťových protokolů a je členěna do dvou hlavních oddílů: teoretického a praktického. Cílem teoretické části práce je provést literární rešerši monografií a elektronických pramenů zabývajících se tematikou síťových protokolů a síťových disků. Dále má rešerše pojednat o datovém úložišti na síti a vytváření velkých souborových systémů.

Hlavním záměrem práce je provést instalaci a následně konfiguraci *Linux*ového serveru, který slouží pro sdílení dat klientům. Dále je důležitým úkolem navrhnout zabezpečení systému a konfigurace správy napájení.

Podstatu textu teoretického oddílu čerpal autor zejména z monografií *Administrace systému Linux* a *Mistrovství ve VMware* a také z internetových zdrojů.

Na počátku teoretické části práce je zařazen nástin problematiky síťových disků. Samotná podstata bakalářské práce je zpracována v oddílech 2- 6, tedy v praktické části. První z těchto kapitol se zaměřuje na instalaci a konfiguraci *NAS* (*Network Attached Storage*) v *Linux*ovém (*Unix*) prostředí. Stěžejní obsah práce je konfigurace protokolů *NFS* (*Network File System*) a *SMB* (*Server Message Block*), implementace podpory *iSCSI* (*Internet Small Computer System Interface*) a návrh zabezpečení disku zpracované v kapitole 3 – 5. Závěrečnou pasáží, neméně podstatnou, je kapitola 6 věnující se power managementu, neboli správě napájení procesoru a disků.

Závěr bakalářské práce shrnuje poznatky, ke kterým jsme došli v jednotlivých podkapitolách a celkově hodnotí problémy samotné instalace a konfigurace.

## **I. TEORETICKÁ ČÁST**

## 1 LITERÁRNÍ ŘEŠERŠE NA TÉMA SÍŤOVÉ DISKY

V dnešní době většina zabezpečovacích systémů firem funguje na základě hardwarové a softwarové komunikace mezi serverem a externími úložišti tzv. *storage*. Existuje však i mnoho jiných možností komunikace zabezpečovacích systémů, které v bezpečnostních systémech hrají velkou roli, ale systém popisovaný v této bakalářské práci, využívá komunikace serveru a externího úložiště.

### 1.1 NAS

*NAS* je zkratka tří anglických slov *Network Attached Storage*, která by se dala volně přeložit jako „datové úložiště na síti“. *NAS* je tedy chápán jako úložiště, které je pomocí různých protokolů připojeno do místní sítě *LAN (Local Area Network)*. *NAS* umí nejen využívat funkce datového serveru, ale může být použit také jako webový nebo *FTP (File Transfer Protocol)* server. Je vybaven interním počítačem, který řídí sdílení dat a podporuje různé síťové protokoly.

*NAS* je vybaven jedním či více pevnými disky, které je možno sloučit do větších datových celků nebo můžou vytvořit *RAID (Redundant Array of Inexpensive/Independent Disks)* pole, které budu popisovat v jedné z následujících kapitol.[1]

#### 1.1.1 Historie

V roce 1980 firma *Newcastle Connection* představila vzdálený přístup k datům přes několik počítačů využívajících operační systém *Unix*. Skupina *Auspex* inženýrů se spojila a v roce 1990 vytvořila *NetApp Filter*, který podporoval jak *Windows CIFS* protokol tak i *Unixový NFS* protokol. Toto odstartovalo výrobu skutečných *NAS*, které známe i dnes.

V roce 2009 začali výrobci představovat možnosti online zálohy dat. Tato funkce měla být implementována přímo do zařízení a měla podporovat online obnovení záloh.[1]



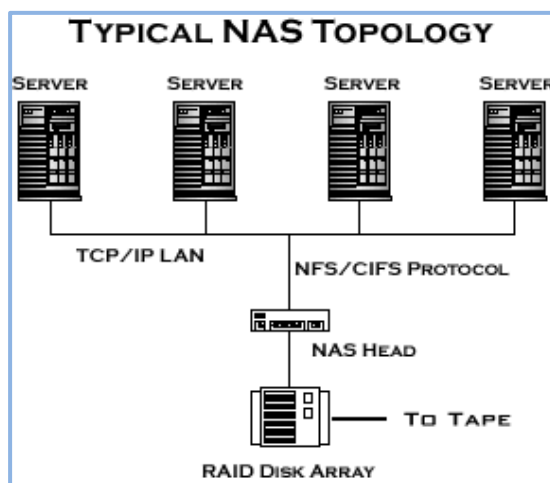
Obr. 1. NAS úložiště [1]

### 1.1.2 Principy NAS

NAS je způsob ukládání dat, který používá speciální zálohovací zařízení s využitím připojení do LAN sítě. Každé zařízení má přidělenou IP (*Internet Protocol*) adresu a tudíž jsou zařízení dostupná klientům přes jiný server, jehož primární funkce je tvořit bránu přístupu k datům. Technologie NAS tedy může být nasazena v prostředích s více servery běžících na různých platformách.

Hlavní výhodou NAS je jeho rozšiřitelnost. Pokud je zapotřebí více diskového prostoru, stačí jen pořídit další NAS zařízení a jednoduše ho připojit pomocí TCP/IP (*Transmission Control Protocol/Internet Protocol*) protokolu do LAN sítě. NAS navíc přináší zabezpečení proti výpadku, to znamená, že data jsou stále dostupná i po výpadku serveru.

NAS zařízení se skládá z NAS hlavy nebo též z NAS boxu, který představuje rozhraní mezi samotným diskovým zařízením a LAN sítí. NAS hlava řídí souborový systém na diskových zařízeních a klient se tedy připojí na NAS hlavu, která má přidělenou IP adresu. Ta převezme požadavek od klienta, zpracuje ho a získá data z diskového pole. Tato data NAS hlava pošle zpět na klientskou stanici. Uživateli se celé NAS jeví jako jeden fyzický disk. Je možné přenášet soubory nebo meta-data souborů (vlastníka, přístupová práva, datum vytvoření, atd.). Hlava sama řídí autorizovaný přístup, zabezpečení a uzamykání souborů.



Obr. 2. Topologie NAS [2]

NAS se z 90 % využívá jako „file storage“ tedy jako datové úložiště. Ve zbylých zhruba 10 % se používá jako jednoduchý web a email server nebo jako levný a jednoduchý rozřazovač serverové zátěže (*load-balancing*).[3]

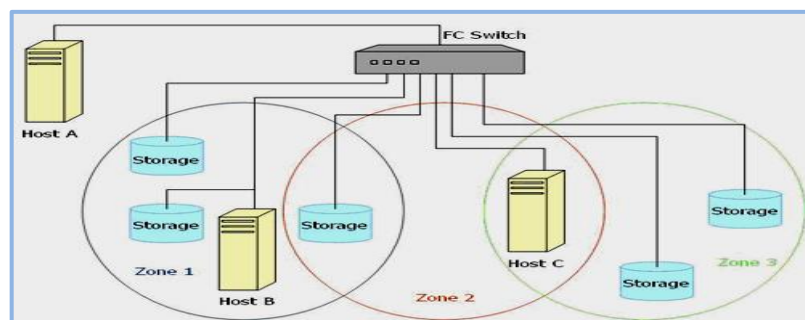
### 1.1.3 NAS versus SAN

Následující podkapitola je zaměřena především na hlavní rozdíly mezi těmito dvěma technologiemi.

NAS poskytuje obojí, jak systém pro ukládání, tak i pro soubory. To se často klade jako kontrast vůči SAN (*Storage Area Network*), který poskytuje pouze systém pro ukládání, který se zakládá na systému blokace a zanechává systém souborů na straně klienta. SAN protokoly jsou *SCSI (Small Computer System Interface)*, *FCP (Fibre Channel Protokol)*, *iSCSI*, *ATA (Advanced Technology Attachment)* přes *Ethernet (AoE)*, nebo *HyperSCSI*. Některé z těchto protokolů budou podrobněji popsány v dalších kapitolách práce.

Jeden způsob, jak jednoduše zobrazit rozdíl mezi NAS a SAN je ten, že NAS se zobrazí do operačního systému klienta jako souborový server, tzn. klient může připojit síťové disky s akcemi na tomto serveru. Zatímco SAN se v operačním systému klienta zobrazuje jakožto disk viditelný ve správě disků *Windows* (společně s lokálními disky klienta) a je možné jej zformátovat souborovým systémem.

I přes rozdíly mezi SAN a NAS, neplatí, že by se navzájem vylučovaly. Mohou být kombinovány v podobě SAN-NAS hybrid s tím, že poskytují ze stejného systému obojí jak protokoly na úrovni souborů (*NAS*), tak i blokované protokoly (*SAN*). Příkladem je *Openfiler*, volný software produkt fungující na *Linuxové* základně.[2]



Obr. 3. Topologie NAS [2]

## 1.2 Výhoda použití RAID

Jednou ze základních technik pomoci, kterých je možno zvýšit bezpečnost ukládaných dat, je použití diskových polí RAID (*Redundant Array of Independent Disks*; původně *Redundant Array of Inexpensive Disks*). Protože požadavky na vysoký výkon a bezpečnost při ukládání dat byly a jsou kritické hlavně na serverech, vzniklo jako první RAID pole pro rozhraní SCSI a až donedávna bylo doménou výkonných a drahých serverů. Nyní se řadiče podporující vytváření RAID polí objevují i pro IDE (*Integrated Drive Electronics*) rozhraní.

Základním principem RAIDu je vytváření jednolitých celků z několika (nejméně dvou) fyzických disků s vyšším výkonem a/nebo bezpečností uložených dat. Původní myšlenkou RAIDu odvozenou z nedostatečných kapacit disků v dobách jeho vzniku je spojováním více malých fyzických disků vytvořit jeden logický s větší kapacitou. Tento způsob se nazývá *spanning* (v překladu přemostění) a v jeho případě se o RAID polí ještě nedá příliš hovořit. Výkon při čtení/zápisu závisí na parametrech disku, na kterém je daný soubor uložen; bezpečnost uložených dat je stejná jako při uložení na jediný disk. Pokud dojde k poškození jednoho z disků, zůstanou data uložená na ostatních discích zachována, nedojde tedy k úplné ztrátě dat.[4]

Celková využitelná kapacita RAID pole se vypočítá pomocí následujícího vzorce:

- C – kapacita nejmenšího použitého disku
- N – celkový počet disků v diskovém poli

Vzorec pro výpočet kapacity: [1]

$$velikost = \frac{c * n}{2}$$

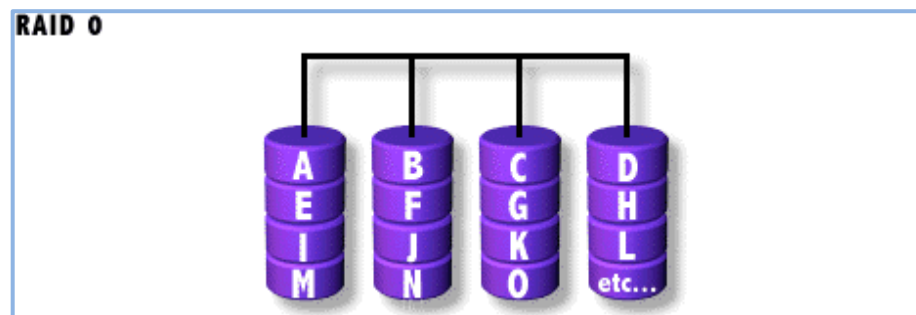
### 1.2.1 Druhy RAIDu

- **RAID 0** - využívá postup známý jako *stripping* neboli zápis po částech. Z několika menších disků vytváří jeden velký, ale každý soubor je rovnoměrně rozložen na všechny disky. To znamená, že při zápisu na disk i při čtení je možné využít maximální přenosovou rychlost nabízenou rozhraním, aniž bychom byli omezeni přenosovou rychlostí jednotlivých disků v poli. V praxi to

funguje tak, že každý soubor, který má být do *RAID 0* pole uložen, je rozdělen na bloky o předem definované velikosti (zpravidla od 1 do 32 kB) a každý blok je uložen na jiný disk. Je-li *RAID* pole vytvořeno ze dvou disků, je první blok souboru uložen na první disk, druhý blok na druhý disk, třetí blok opět na první disk atd.[6]

Výhodou *RAID 0* je vyšší rychlost diskového subsystému, bezpečnost dat ovšem není nijak zvýšena. Pokud dojde k havárii jednoho z disků, jsou ztracena všechna data.

Pro *RAID 0* je nejvhodnější použít disky o stejné kapacitě. Protože soubory jsou ukládány rovnoměrně na všechny disky, je celková kapacita *RAID 0* pole dána kapacitou nejmenšího z disků znásobenou počtem disků. Jsou-li použity dva disky, jeden s kapacitou 6 GB a druhý s kapacitou 8 GB, bude celková kapacita *RAID 0* pole 12 GB a zbývající 2 GB na větším disku nebudou využity. Z hlediska výkonu je vhodné použít disky se stejnou rychlostí, nejlépe pak zcela identické.[6]



Obr. 4. RAID 0 [7]

- **RAID 1** - pracuje s technikou nazvanou *mirroring* neboli zrcadlení. Zvyšuje bezpečnost uložených dat tím, že data ukládá současně na všechny disky v poli, vytváří „zrcadlový obraz“ disků. V případě havárie jednoho z disků v *RAID 1* poli jej zcela zastoupí další disk v poli, aniž by bylo nutné data jakkoli kopírovat nebo dopočítávat, protože jsou uložena vícekrát na několika discích.

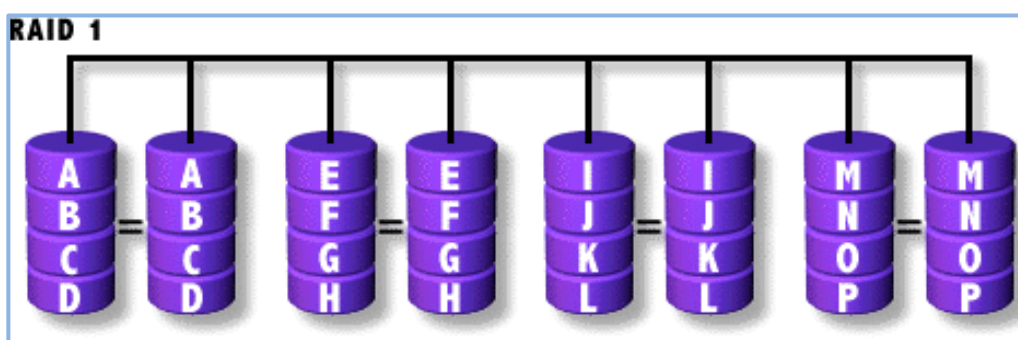
I pro *RAID 1* je nejvhodnější použít disky o stejné kapacitě, celková kapacita *RAID* pole je dána kapacitou nejmenšího disku v poli. Z 6 GB a 8 GB

disků tedy lze vytvořit *RAID 1* pole s kapacitou 6 GB, zbývající 2 GB na větším disku budou opět nevyužity. I zde je nevhodnější použít identické disky, neboť jejich výkon je dán výkonem nejpomalejšího z nich.

Výhodou *RAID 1* je zajištění bezpečnosti dat jejich duplikací, k nárůstu výkonu diskového subsystému však nedochází.[5, 8]

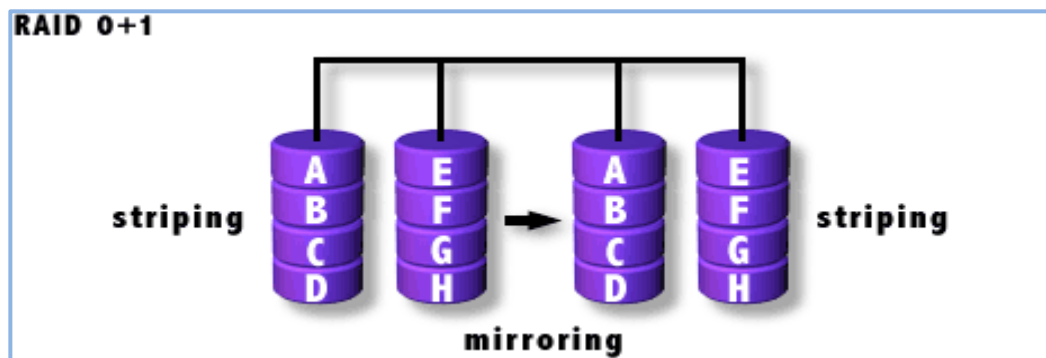
### Práce s daty v poli *mirror* disků:

- 1) Zapisuje-li se na zrcadlové disky nějaký soubor, řadič souběžně zapisuje identické kopie souboru na každou mechaniku v poli. Pole zrcadlových disků tedy musí mít alespoň dva disky.
- 2) Pokud se soubor z diskového pole čte, řadič střídavě čte skupiny sektorů z každé mechaniky a z kousků souboru sestavuje celky dodávané do počítače. Tento proces čtení zrychluje. Rychlost závisí na počtu mechanik v poli. Když jsou zrcadleny dvě mechaniky, zkracuje se čas potřebný pro čtení přibližně na polovinu; tři zrcadlové disky zkracují čas čtení asi na třetinu vůči čtení z disku jediného.
- 3) V případě chyby při čtení – způsobené, buď defektem na povrchu jedné mechaniky, nebo havárií celého jednoho disku řadič jednoduše přečte neporušenou část souboru z druhého disku.
- 4) Jestliže je chyba čtení způsobena poškozením média, řadič automaticky přečte data z kopie souboru na druhé mechanice a zapíše je do nové nepoškozené oblasti na mechanice, kde chyba nastala.[7]



Obr. 5. RAID 1 [7]

**RAID 0 + 1 - Striping / mirroring** je kombinací obou předchozích typů, data jsou rozdělována do bloků a takto vytvořené **RAID 0** pole je zrcadleno. Data jsou tedy uložena několikrát (nejčastěji dvakrát) a přitom každý obraz je sám **RAID 0** polem. **RAID 0+1** tedy zvyšuje výkon diskového subsystému i bezpečnost uložených dat. Je však nutné použít nejméně čtyři pevné disky, opět nejlépe identické.[7]



Obr. 6. RAID 0+1 [7]

### 1.2.2 Paritní RAID (RAID 5, RAID 6)

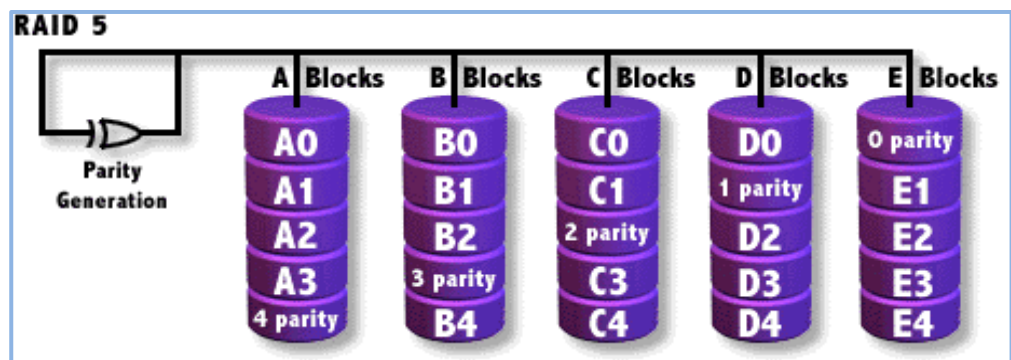
Tato úroveň **RAIDu** využívá k reprezentaci dat na několika discích matematický výpočet (výpočet parity pomocí *XOR*). Ukazuje se, že jde o přijatelný kompromis mezi dostupností **RAID 1** a kapacitou **RAID 0**. **RAID 5** počítá paritu přes všechny mechaniky v dané množině a zapisuje ji na jinou mechaniku. Výpočet a zápis paritního bloku probíhá v rámci polí **RAID 5** cyklicky. Zde je důležité uvést **RAID 4**, který je variantou, kdy parita nerotuje po mechanikách, nýbrž se zapisuje na jednoúčelový paritní disk.

Schémata paritního **RAIDu** poskytují velmi dobrý výkon. Existuje zde však i určitá forma „daně ze zápisu“. Při zápisu celého pruhu je touto daní výpočet a zápis parity, avšak nezapisuje-li se jenom část pruhu, musí dojít k opakovanému přečtení starého obsahu. Je tedy nutné jej přečíst, vypočítat novou paritu a starý blok aktualizovat. Naopak rychlost čtení je vynikající, neboť od zrcadlených schémat **RAIDu** lze číst z mnoha mechanik.[9]

- **RAID 5** - V současné době asi nejpoužívanější **RAID 5** také používá techniku disk *striping* a zapisovaná data rozkládá na jednotlivé disky po celých diskových sektorech, podobně jako **RAID 4**. Nepoužívá však vyhrazený paritní

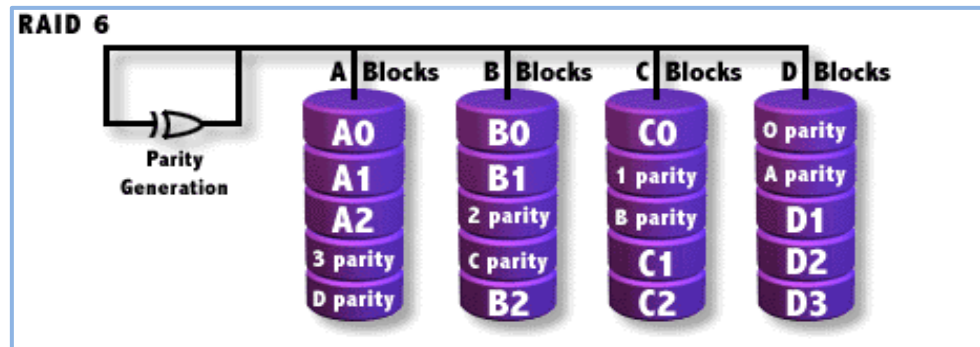
disk, ale místo toho příslušné zabezpečovací informace rozkládá rovnoměrně po všech discích, které jsou v poli k dispozici. Tento typ diskového pole může být volitelně rozšířen o tzv. *stand-by disk*, tj. fyzický disk, který v nečinnosti čeká na výpadek některého aktivního disku v poli, aby jej (po proběhnutí synchronizace) mohl beze ztráty dat nahradit.

*RAID 5* nabízí jak odolnost vůči výpadku (jednoho ze svých disků), tak i celkové zrychlení při práci s disky. Vzhledem k režii, která připadá na práci s paritními daty, je tento typ diskového pole (stejně jako předchozí dva typy) o něco pomalejší než *RAID 0*. [7]



Obr. 7. RAID 5 [7]

*RAID 6* - vychází z *RAID 5*, oproti němu však obsahuje druhé (nezávislé) paritní schéma, které je rozloženo na jiných discích, než schéma primární. Díky tomu poskytuje tento typ diskového pole extrémně vysokou ochranu dat při selhání některého z disků. *RAID 6* je díky tomu spolehlivější a i při výpadku dvou disků lze data znovu zrekonstruovat. Rychlost čtení je srovnatelná s *RAID 5*, avšak zápis je o něco pomalejší, protože je nutné vypočítat a uložit paritní informace dvakrát. Také cena *RAID 6* pole je o něco vyšší. Proto se používá jen tam, kde je kladen opravdu důraz na spolehlivost a přístupnost dat i přes malou pravděpodobnost výpadku dvou disků najednou. [5]



Obr. 8. RAID 6 [7]

### 1.3 LVM

*LVM (Logical Volume Management)* je v *Linuxu* nástroj pro spojení logických zařízení do větších celků. Využívá se pro vytváření velkých souborových systémů, pro možnost plynulého zvyšování kapacity souborových systémů bez nutnosti kopírování dat, pro možnost vyměňování disků, vytváření snímků souborových systémů a podobně.

*LVM* operuje s *PV (Physical Volume)*, *VG (Volume Group)* a *LV (Logical Volume)*. Základem *LVM* jsou bloková zařízení (*PV*), ze kterých se pak sestavují svazky (*VG*) a na nich se vytvářejí logické jednotky (*LV*), které vystupují jako bloková zařízení a lze je tedy snadno naformátovat a použít jako nosné médium pro souborový systém.[10,11]

#### 1.3.1 Fyzický svazek

Fyzickým svazkem, který je označován jako *PV*, jsou bloková zařízení (nejčastěji oddíl, ale může to být třeba i celý disk, případně *RAID* pole). *PV* se vytváří příkazem `pvcreate /dev/sda3` a dostupné *PV* lze vypsát příkazem `pvscan` nebo `pvdisplay`. [10]

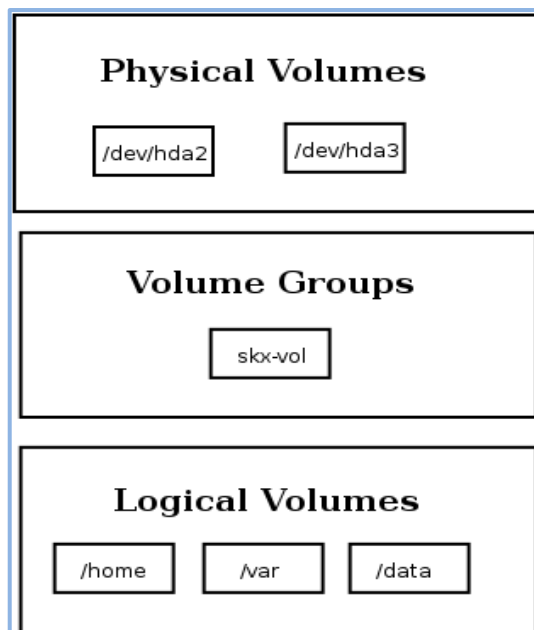
#### 1.3.2 Skupina svazků

Skupina svazků označována jako *VG (Volume Group)* sdružuje svazky *Physical Volume* do větších celků. Parametrem příkazu `vgcreate` je název nového svazku a pak seznam *PV*, které budou do *VG* zařazeny. Další *PV* lze samozřejmě do svazku přidávat

dodatečně, případně je i odebrat (vše za provozu). Příkaz pro vytvoření svazku je: `vgcreate NÁZEV_VG /dev/sda3`. Pokud je nutné vyhnout se restartu systému, je nezbytné VG aktivovat příkazem `vgchange -a y`. [10]

### 1.3.3 Logická jednotka

Logická jednotka, označována jako *LV*, vytvoří na *Volume Group* logickou jednotku, se kterou se dále může pracovat jako s běžným blokovým zařízením (např. oddílem na pevném disku). Následující příkaz vytvoří 1 GB velkou logickou jednotku: `lvcreate -L10000 -n NÁZEV_LV NÁZEV_VG`. [10]



Obr. 9. Skladba LVM [12]

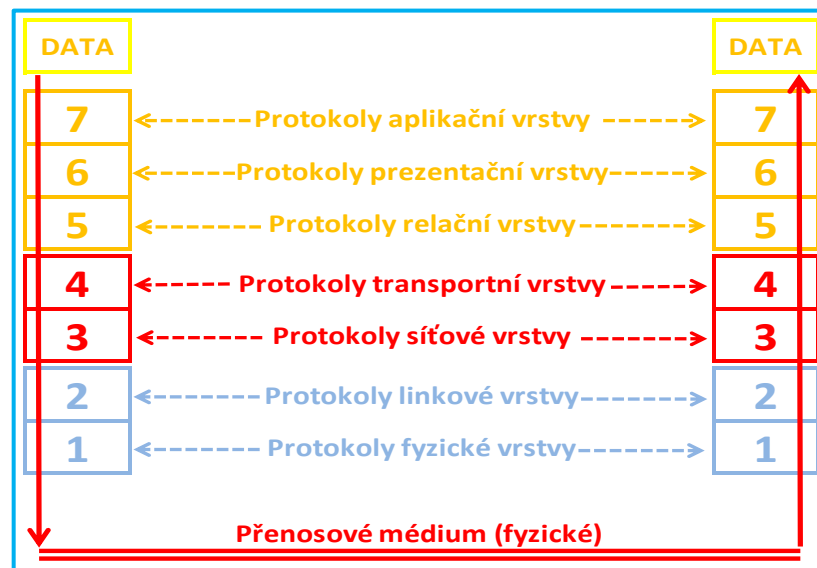
## 1.4 Síťové protokoly

Protokol jako takový je v informatice standart podle, kterého probíhá elektronická komunikace a přenos dat mezi dvěma koncovými body. V té nejjednodušší podobě protokol definuje syntaxi, sémantiku a synchronizaci vzájemné komunikace. Mohou být realizovány hardwarově, softwarově anebo kombinovaně.

Protokoly mohou specifikovat mnoho vlastností, jako jsou například:

- detekce základního fyzického spojení (metalické, optické, bezdrátové), existence jiných koncových bodů nebo uzlů
- vyjednávání o různých parametrech spojení
- jak začít a ukončit spojení
- co dělat, když dojde k poškození nebo k nesprávnému naformátování dat, atd.

Protokoly pracují na různých vrstvách modelu *ISO/OSI* a *TCP/IP* modelu. V každé vrstvě *OSI (Open Systems Interconnection)* je přiřazena sada protokolů, které plní její funkce. Tyto protokoly komunikují se stejnými protokoly protilehlého systému - tzv. horizontální komunikace pomocí datových celků *PDU (Protocol Data Unit)*. [13,14]



Obr. 10. Vrstvy modelu OSI [13]

*TCP/IP (Transmission Control Protocol/Internet Protocol)* byl vytvořen roku 1970 agenturou *DARPA (Defense Advanced Research Projects Agency)*, v letech 1977-1979

probíhal jeho vývoj pro *UNIXy*, 1983 byl implementován do sítě *ARPANET T Internet*. Je také někdy nazýván jako *Internet Reference Model*, *DoD Model (Department of Defense)* nebo *ARPANET Reference Mode* a pokrývá SW 2. až 7. vrstvu modelu *OSI*, u 2. vrstvy jen podvrstvy *LLC (Logical Link Control)*. [13,14,15]



Obr. 11. Rozdíl mezi vrstvami OSI a TCP/IP modelu [13]

#### 1.4.1 Vrstvy TCP/IP modelu a příklady síťových protokolů

1) **Aplikační vrstva** – je nejvyšší vrstvou *TCP/IP* modelu. Protokoly této vrstvy specifikují pravidla komunikace a formáty datových struktur pro jednotlivé síťové služby. Některé služby jsou vázány na konkrétní komunikační protokol, jiné mohou volit mezi *TCP* a *UDP*. Na této vrstvě pracují následující protokoly:

- **FTP – (File Transfer Protocol)** je v informatice protokol pro přenos souborů mezi počítači pomocí počítačové sítě. Využívá protokol *TCP* z rodiny *TCP/IP* a může být aplikován nezávisle na použitém operačním systému.
- **SMTP – (Simple Mail Transfer Protocol)** je internetový protokol určený pro přenos zpráv elektronické pošty (e-mailů) mezi přepravci elektronické pošty (*MTA*). Protokol zajišťuje doručení pošty pomocí přímého spojení

mezi odesílatelem a adresátem; zpráva je doručena do tzv. poštovní schránky adresáta, ke které potom může uživatel kdykoli přistupovat (číst zprávy) pomocí protokolů *POP3* nebo *IMAP*.

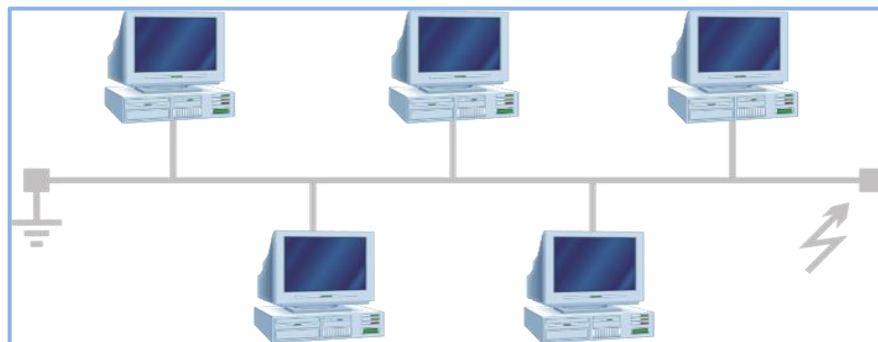
- *SMB a NFS* – tyto protokoly budou podrobněji popsány v další části.

**2) Transportní vrstva** - umožňuje adresovat přímo aplikace (například v protokolech *TCP/IP* pomocí čísel portů). Poskytuje transparentní, spolehlivý přenos dat s požadovanou kvalitou. Vyrovnává různé vlastnosti a kvalitu přenosových sítí. Provádí převod transportních adres na síťové, ale nestará se o směrování.

- *TCP (Transmission Control Protocol)* - je jedním ze základních protokolů sady protokolů Internetu, konkrétně představuje transportní vrstvu. Použitím *TCP* mohou aplikace na počítačích propojených do sítě vytvořit mezi sebou spojení, přes které mohou přenášet data. Protokol garantuje spolehlivé doručování a doručování ve správném pořadí. *TCP* také rozlišuje data pro vícenásobné, současně běžící aplikace (například webový server a emailový server) fungující na stejném počítači.
- *UDP (User Datagram Protocol)* - je jedním ze sady protokolů internetu. O protokolu *UDP* říkáme, že nedává záruky na datagramy, které přenáší mezi počítači v síti. Někdy je označován jako „nespolehlivý“, ale to je velmi zavádějící označení. Na rozdíl od protokolu *TCP* totiž nezaručuje, zda se přenášený datagram neztratí, jestli se nezmění pořadí doručených datagramů nebo zda se některý datagram nedoručí vícekrát. Je však typický pro aplikace „dotaz/odpověď“ nevyžadující vysoké zabezpečení a aplikace požadující jednoduchost.

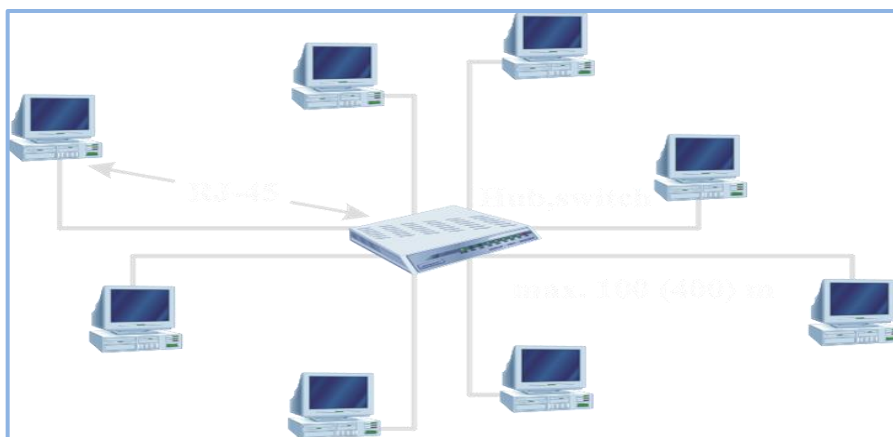
**3) Internetová (síťová) vrstva** - tato vrstva se stará o směrování v síti a síťové adresování. Poskytuje spojení mezi systémy, které spolu přímo nesousedí. Obsahuje funkce, které umožňují překlenout rozdílné vlastnosti technologií v přenosových sítích

- **IPV4 (Internet Protocol version 4)** – vytváří základ pro komunikaci v rámci sítě Internet. *IPV4* datagram nese pouze informaci o kontrolním součtu hlavičky datagramu se služebními údaji.
  - **ICMP (Internet Control Message Protocol)** – počítače v síti tento protokol používají pro odesílání chybových zpráv, například že požadovaná zpráva není dostupná.
- 4) **Linková vrstva** – poskytuje spojení mezi dvěma nebo více systémy zapojených na stejné lince. Seřazuje přenášené rámce, stará se o nastavení parametrů přenosu linky, detekuje neopravitelné chyby.
- **ETHERNET** – v současné době nejrozšířenější technologie pro budování počítačových sítí. *Ethernet* a jeho síťové rozhraní pracuje pouze s tak zvanými „ethernetovými rámci“.
  - **FDDI (Fiber Distributed Data Interface)** - protokol určuje přístup k přenosovému zařízení a ověřuje validitu doručených dat.
- 5) **Fyzická vrstva** - přenáší zakódovanou nebo modulovanou posloupnost bitů mezi dvěma přímými sousedy prostřednictvím komunikačního média v *LAN* např. *PC (Personal Computer) - PC, PC - hub, PC - switch, switch - switch, PC - router, router – router*. Zabývá se elektrickými a mechanickými vlastnostmi přenosového média.
- **10 BASE2** – standart sítě *ethernet* založený na tenkém koaxialním kabelu s rychlostí přenosu 10Mbit/s, používá sběrníkovou topologii.[15]



Obr. 12. Příklad sběrníkové topologie [13]

- **10 BASE2T**- každý uzel je připojen k hubu nebo na *switch*, který plní roli společného přenosového média (slouží jako přenosová stanice), jako přenosové médium se používá kroucená dvojlinka a přenosová rychlost je 10 Mb/s.



Obr. 13. Příklad sítě 10Base2T [13]

#### 1.4.2 Síťový protokol *SMB* – (*SAMBA* /*CIFS*)

*SMB* je zkratka z anglického slova *Server Message Block*, jedná se o síťový protokol, který figuruje na aplikační vrstvě modelu *OSI*. Slouží ke sdílenému přístupu k souborům, tiskárnám, sériovým portům a k další komunikaci mezi uzly na síti. Je hlavně používán v počítačích s operačním systémem *Windows*. Tento protokol také poskytuje autentizovaný mechanismus pro procesorovou komunikaci.

Protokol *SMB* vyvinul ve firmě *IBM* (*International Business Machines Corporation*) Barry Feigenbaum a jeho cílem bylo změnit *DOS*ový přístup k lokálním souborům na síťový systém souborů. *Microsoft* provedl implementaci protokolu do produktu *LAN Manager*, který vyvíjel spolu s firmou *3Com* přibližně v roce 1990.

Díky firmě *Microsoft* je protokol rozšířený v oblasti komunikace mezi uživateli v *LAN* síti a souborovými servery. Protokol *SMB* využívá souborové a tiskové servery síťových operačních systémů *IBM* a *Microsoft*. *SMB* pracuje na principu klient-server. Server vlastně umožňuje klientům síť přistupovat ke sdíleným prostředkům, například (sdílené disky, adresáře, tiskové fronty). Veškeré síťové prostředky jsou rozpoznávány pomocí *UNC* (*Uniform Naming Convention*)

– síťové adresy. *UNC* adresa se udává ve tvaru `\\jméno_serveru\jméno_zdroje`. Klientská část má za úkol definovat požadavky na sdílené prostředky serveru. Server vykoná rozbor požadavků odeslaných klientem ve tvaru paketu *SMB*, porovná přístupová práva a na základě přístupových práv zahájí požadovanou operaci (přejmenování, smazání souboru nebo adresáře, vytvoření souboru nebo adresáře, otevření souboru). Výsledek je klientovi poslán s odpovědí identickým blokem *SMB*. [11,16]

#### **Dva pohledy *SMB* serveru k přístupu ke sdíleným prostředkům:**

- 1) *SHARE LEVEL* – přístup z pohledu na úrovni sdíleného prostředku
  - Server povoluje přístup ke sdíleným prostředkům na základě správného hesla, které je přiřazeno k jednotlivým sdíleným prostředkům. Pokud uživatel zadá správné heslo, pak je klientovi přidělen identifikátor prostředku *network ID* pomocí, kterého přistupuje k prostředkům.
- 2) *USER LEVEL* – přístup z pohledu na uživatelskou úroveň
  - Uživatel se přihlašuje na server okamžitě pomocí jména a hesla. Pokud jsou logovací údaje správné, tak server přidělí uživateli *user ID*, pomocí něhož server odvozuje přístupová práva.

Komunikace mezi klientem a serverem probíhá tak, že uživatel zašle serveru požadavek a následně dochází ke specifikaci parametrů spojení a verzí protokolů mezi klientem a serverem. Klient odešle požadavek, kde je uvedeno jméno a heslo uživatele, a pokud je server v režimu *USER LEVEL*, tak přidělí uživateli *user ID*. [11,16,17]

### **1.4.3 Síťový protokol *NFS***

Síťový protokol *NFS* (*Network File System*) je vlastně internetový protokol pro vzdálený přístup k souborům přes počítačovou síť. Funguje především nad transportním protokolem *UDP*, ale od verze 3 je možné ho porovnat také nad protokolem *TCP*. Jeho použití by se dalo charakterizovat tak, že jeho

prostřednictvím může klient připojit disk ze vzdáleného serveru a s ním pracovat jako s lokálním. V prostředí *Linux* je to nejpoužívanější protokol pro tyto účely.

Za kolegu *NFS* lze považovat protokol *NIS* (*Network Information Service*), který slouží k distribuci nejrůznějších konfiguračních dat v počítačové síti.

*NFS* přispívá k transparentnosti sítě, protože umožňuje připojovat adresářové struktury z jiných hostitelů, jako by to byly lokální souborové systémy, takže pak vypadají jako normální adresáře uživatelů na lokálním počítači. Například všechny domovské adresáře uživatelů mohou být na centrálním serveru, ze kterého si je připojují všichni v lokální síti. V důsledku toho se uživatelé mohou přihlásit na libovolném počítači a získají vždy stejný domovský adresář. Podobně je možné sdílet velké objemy dat (například databáze, dokumentaci, atd.) mezi mnoha počítači tak, že jediná kopie dat bude udržovaná na serveru a ostatní počítače k ní budou mít přístup.[11,18]

#### 1.4.3.1 Konfigurace NFS serveru

*NFS* server se edituje pomocí konfiguračního souboru `/etc/exports`, který na jednotlivých řádcích obsahuje definice sdílených adresářů. Jako první je název adresáře a pak seznam povolených klientů (zde jsou uvedeny názvy server, stanice a *IP* adresa) s přidáními volitelnými parametry:[11,18]

- `/usr 192.168.57.124(ro) stanice(ro)`
- `/home 192.168.57.124(rw, no_ROOT _squash) stanice(rw)`

##### Popis parametrů:

- `ro` (read only) – pouze pro čtení
- `rw` (read and write) – povoleno čtení i zápis
- `no_ROOT _squash` – mapovat požadavky na běžného uživatele

#### 1.4.3.2 Konfigurace klienta

Klient připojuje adresář ze serveru do svého adresářového stromu stejným způsobem, jako jsou připojovány jednotlivé systémy souborů. Je nutné na klientské stanici spustit též `portmap`:

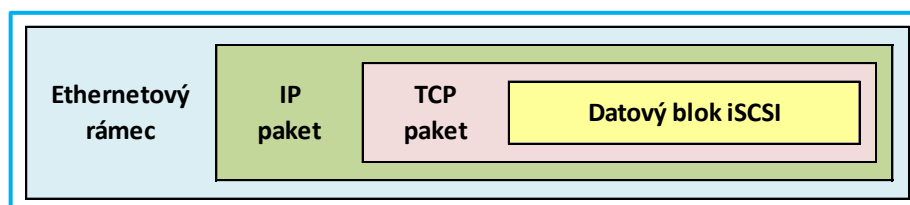
- `mount -t NFS server:/home/home`
- `mount -t NFS server:/usr/mnt/usr-from-server`

Od této chvíle může uživatel s daty v adresáři `/home` a `/mnt/usr-from-server` pracovat, jako by byly uloženy na lokálním disku.[11]

#### 1.4.4 Protokol iSCSI

Koncepce *iSCSI* (*Internet Small Computer System Interface*) vychází ze dvou technologií. *SCSI* rozhraní pro připojování disků v serverech a rodiny protokolů *TCP/IP*. Z rozhraní *SCSI* se používá pouze protokol, kterým spolu zařízení komunikují a zcela opouští jeho fyzickou vrstvu (kabely, konektory, elektrickou specifikaci). Pro přenos paketů *SCSI* se použije jejich zapouzdření do rodiny protokolů *TCP/IP*.

*iSCSI* je norma *IETF* (*Internet Engineering Task Force*) pro zapouzdření zařízení a dat *iSCSI* do paketů *TCP/IP*, které jsou naopak zapouzdřeny v *ethernetových* rámcích. Na obr. 13 je znázorněno zapouzdření *iSCSI* v *TCP/IP* a v *ethernetových* rámcích.[9]



Obr. 14. Zapouzdření *iSCSI* v *TCP/IP* a *ethernetových* rámcích [9]

Opakovaný přenos *TCP* je určen pro zpracování ztracených *ethernetových* rámců nebo vážných přenosových chyb. Provoz úložiště je vzhledem k běžnému provozu v *LAN* (*Local Area Network* – Místní síť) poměrně intenzivní. Tím se minimalizují opakované přenosy, ztracené rámce a je tak získána spolehlivá *ethernetová* infrastruktura, která je pro *iSCSI* velmi důležitá.

#### 1.4.4.1 Bezpečnost iSCSI

Vysokou roli u *iSCSI* hraje bezpečnost přenášených dat. *iSCSI* pakety jsou posílány přes firemní síť *LAN* nebo přes internet, a proto musejí být maximálně chráněny proti útokům založených na zfalšování identity *iSCSI Initiatoru* (tzv. *spoofing*) a odposlouchávání (tzv. *man in the middle*). Proti *spoofingu iSCSI Initiatoru* chrání *iSCSI* autentizace založená na *CHAP* (*Challenge Handshake Authentication Protocol*). Proti odposlouchávání *iSCSI* chrání protokol *IPSec* (*Internet Protocol Security*). Ten buď *iSCSI* pakety pouze podepisuje (detekce odposlouchávání a ochrana proti modifikaci), nebo kompletně šifruje (nikdo nemůže tato data odposlouchávat ani modifikovat).[19]

- **CHAP** – je často používaný autentifikační protokol, v němž se k autentifikaci zdroje a cíle komunikace používá výměna hesel. Nepřímý *CHAP* je jednocestný; zdroj se autentizuje cíli, anebo, v případě *iSCSI*, ještě před zahájením komunikace se spouštěč *iSCSI* autentizuje cíli *iSCSI* a naopak. Vzhledem k tomu, že úložiště a obecný provoz po síti mohou sdílet síťovou infrastrukturu, *CHAP* je volitelný mechanismus k autentifikaci zdroje a cíle pro zvýšení bezpečnosti v rámci *iSCSI*.
- **IPsec** – je norma *IETF*, která pro zabezpečení obsahu *iSCSI* proti útokům prostředníka využívá k šifrování principy veřejného klíče. Podobně, jako je tomu i v případě autentizace *CHAP*, i tento volitelný vyšší stupeň zabezpečení je součástí normy *iSCSI*, neboť pro přenosy *iSCSI* je možné využívat univerzální síť *IP* a v tomto případě představují nezašifrovaná data bezpečnostní riziko. Přesto se používá *IPsec* poměrně zřídka, neboť značně zatěžuje *CPU* (*Central Processing Unit*) spouštěče i cíle.[19]

## **II. PRAKTICKÁ ČÁST**

## 2 INSTALACE A KONFIGURACE NAS V LINUXOVÉM (UNIX) PROSTŘEDÍ

Hlavním úkolem první části je provést vlastní instalaci a konfiguraci serveru v *Linux*ovém prostředí. Tato část práce popisuje hardwarovou konfiguraci použitého serveru, instalaci operačního systému *Linux DEBIAN* a konfiguraci disků pro *NAS*.

### 2.1 HW konfigurace použitého serveru

Pro instalaci a veškerou konfiguraci je použito šasi serveru od společnosti Supermicro typ 5017C-MTF. Tento server byl zapůjčen UTB (Univerzita Tomáše Bati).

**Server je osazen:**

- Intel Atom CPU D525 1.80 GHz
- Základní deska GIGABYTE GA D525TUD
- Pevné disky: - SEAGATE ST3120026AS 120 GB  
- SAMSUNG SP1614C 160 GB
- Operační paměť KINGSTON K2 2x4 GB
- Zdroj ABLECOM SP302-1S 300 W

Z použité hardwarové konfigurace je zřejmé, že se nejedná o server s vysokým počtovým výkonem, nicméně pro stabilní chod *Linux*ového datového serveru je plně dostačující.

### 2.2 Použitý operační systém

Pro tuto bakalářskou práci je zvolen operační systém *Linux DEBIAN* verze 6.0.7-i386. Tato konkrétní verze *Linux*ové distribuce *Debian* je záměrně vybrána z důvodu její stability v serverovém prostředí a samozřejmě i pro její jednoduchost údržby pro správce informačních systémů.

*Debian* není vyvíjen komerčním subjektem, ale řadou dobrovolníků z celého světa. Je vydáván ve třech verzích:

- 1) Verze „*stable*“ – jedná se o stabilní, pečlivě otestovaný a chyb zbavený software, připravený pro nasazení i v kritických aplikacích. Vzhledem k tomu, že tato distribuce je vždy poněkud zastaralá, jsou pro ni vydávány „záplaty“ řešící bezpečnostní problémy a kritické chyby. Z důvodu zastaralosti softwaru se příliš nepoužívá v desktopech.
- 2) Verze „*testing*“ – testovací verze s novějším softwarem, ale s možným výskytem chyb.
- 3) Verze „*unstable*“ – nestabilní verze, vývojářská část, používaná převážně vývojáři. Nejedná se však o nestabilní vydání, obsahuje pouze novější software, který nebývá odladěn.[22]

Tato *Linuxová* distribuce *Debian* je považována za nejvhodnější pro méně náročné společnosti. Tedy nulové cenové náklady na pořízení licencí a také její nízké hardwarové požadavky jsou její značnou výhodou.

Jsou samozřejmě vydávány i jiné *Linuxové* distribuce jako *Fedora*, *Ubuntu*, *Mandriva*, atd. nicméně jejich použití je především v desktopech pro jejich propracované *x-window* prostředí, které je pro uživatele příjemnější než u distribuce *Debian*.

## 2.3 Postup instalace

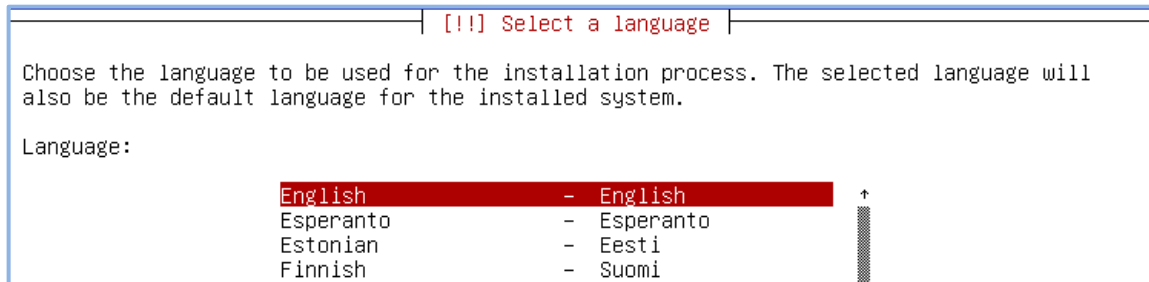
Možnosti instalace jsou dány hardwarovým vybavením serveru. Jednou z možností instalace je nabootováním z *USB Flash* disku. Důvodem pro tento konkrétní způsob je, že šasí použitého serveru není vybaveno *CD (Compact Disc)* mechanikou.

### 2.3.1 BOOT

Stiskem funkční klávesy F12 se otevře okno v *BIOSu (Basic Input-Output System)* umožňující volbu zdroje *BOOT*. Výběrem položky *HDD-USB* dojde k *nabootování* instalace operačního systému z *USB Flash* disku.

### 2.3.2 Volba jazyka instalace

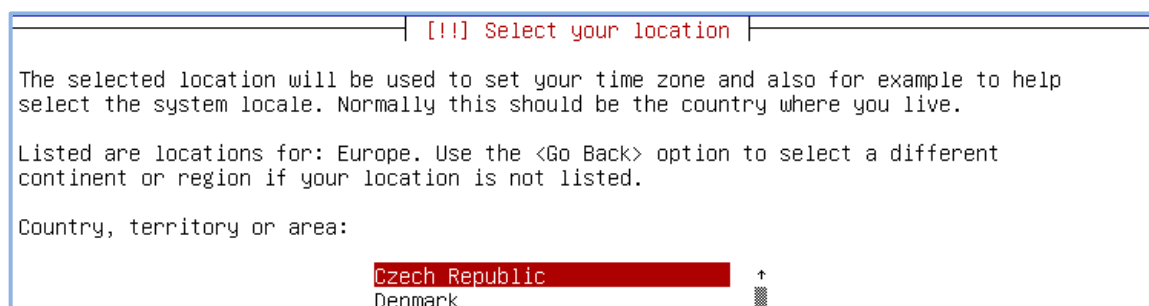
Výběr jazyka instalace je zásadní pro další kroky instalace a zvolený jazyk je použit jako výchozí i v nainstalovaném systému. Vzhledem k tomu, že angličtina je mezinárodní jazyk celosvětově používaný v informačních systémech, je zvolena i pro tuto instalaci.



Obr. 15. Výběr jazyka instalace

### 2.3.3 Výběr místa

Ačkoliv položka výběr místa se může zdát nepodstatná, tak ve skutečnosti výběr místa, tedy země, má vliv na časové pásmo, ve kterém se server nachází. Čas serveru se samozřejmě dá upravovat i později pomocí příkazu `date mmddhhMMYYYY`. Nicméně výrazně pohodlnější je automatická synchronizace s *NTP* serverem pomocí příkazu `ntpdate server`. Ze zmíněného důvodu je zvolena lokalizace Česká republika.



Obr. 16. Volba časového pásma

### 2.3.4 Rozdělení disků

Rozdělení disků je stěžejní částí instalace a může být rozdělena do tří zásadních kroků:

1. **TYP RAIDu** - nejprve je důležité zvolit typ *RAIDu* mezi disky. Pro konkrétní instalaci je zvolen typ *RAID1*, který může být považován za nejvhodnější pro datový server se dvěma pevnými disky. Rychlost zápisu a čtení není tak vysoká jako u *RAID 0*, ale při výpadku jednoho z disků u *RAID 1* nedochází ke ztrátě uložených dat. Proto pro konfiguraci *NAS* je zvolen *RAID 1*.

```

[!!!] Partition disks

This is an overview of your currently configured partitions and mount points. Select a
partition to modify its settings (file system, mount point, etc.), a free space to create
partitions, or a device to initialize its partition table.

Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes

RAID1 device #0 - 21.5 GB Software RAID device
#1          21.5 GB          57.9 kB          unusable
SCSI3 (0,0,0) (sda) - 21.5 GB VMware, VMware Virtual S
#1 primary 21.5 GB          K raid
SCSI3 (0,1,0) (sdb) - 21.5 GB VMware, VMware Virtual S
#1 primary 21.5 GB          K raid

Undo changes to partitions
Finish partitioning and write changes to disk

```

Obr. 17. Konfigurace RAID 1

2. **VOLUME GROUP** – vytvořený *RAID device* je nutné převést na *volume group*. Uvnitř *volume groupy* je pak možno vytvářet logické svazky, u kterých je možno navyšovat diskový prostor, případně jej zmenšit.

```

[!!!] Partition disks

This is an overview of your currently configured partitions and mount points. Select a
partition to modify its settings (file system, mount point, etc.), a free space to create
partitions, or a device to initialize its partition table.

Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes

RAID1 device #0 - 21.5 GB Software RAID device
#1          21.5 GB          57.9 kB          unusable
SCSI3 (0,0,0) (sda) - 21.5 GB VMware, VMware Virtual S
#1 primary 21.5 GB          K raid
SCSI3 (0,1,0) (sdb) - 21.5 GB VMware, VMware Virtual S
#1 primary 21.5 GB          K raid

```

Obr. 18. Vytvoření volume groupy

Na obr. 16 je možné vidět vytvořený *RAID 1* se zařízeními *sda* a *sdb*. Poté z *RAID* byla vytvořena *volume group*.

**3. LOGICKÉ VOLUME** – nyní je možné z vytvořené *volume group* vytvořit *logické volume* a jim přiřadit diskový prostor. Na obr. 17 je nyní vidět, jaké rozdělení je vybráno. Tedy pro adresář *ROOT* je zvolen 1 GB (*Giga Byte*) prostor naformátovaný na souborový systém *ext2*. Pro adresář *data*, který bude při instalaci vytvořen, je zvolen diskový prostor 10 GB a souborový systém *ext3*. Na konec je zvolen 1 GB prostor pro adresář *swap*, který systém používá jako virtuální paměť. Zbývající prostor 8 GB bude použit dále pro navyšování diskového prostoru pro adresář *data*, která budu provádět již pomocí příkazové řádky.

```
LVM VG source, LV data - 10.0 GB Linux device-mapper (linear)
#1          10.0 GB  f  ext3      /data
LVM VG source, LV root - 998.2 MB Linux device-mapper (linear)
#1          998.2 MB  F  ext2      /
LVM VG source, LV swap - 998.2 MB Linux device-mapper (linear)
#1          998.2 MB  f  swap      swap
RAID1 device #0 - 21.5 GB Software RAID device
#1          21.5 GB   K  lvm
           57.9 kB   unusable
```

Obr. 19. Logické volume

### 2.3.5 Dokončení instalace

Po nastavení diskových oddílů systém pokračuje zápisem změn na disk. Dále je možno provést volbu části systému požadované pro konkrétní požadavky instalace. Při konfiguraci *NAS* se provádí volba datového serveru a *SSH (Secure Shell)* serveru. *SSH* server je zvolen z důvodu možného terminálového přístupu k serveru, čili pro možnost jeho vzdálené správy.

### 2.3.6 Kontrola instalace

Jakmile se dokončí instalace, je možné přihlášením do systému ověřit, zda instalátor správně přiřadil diskový prostor a zda je spuštěna služba *SSH* serveru. Zobrazení informací o *volume group* je realizováno pomocí příkazu `vgdisplay -v`, kde parametr `-v` zobrazí podrobné informace:

```
root@debian:~# vgdisplay -v
  Finding all volume groups
  Finding volume group "source"
  --- Volume group ---
  VG Name                source
  System ID
  Format                  lvm2
  Metadata Areas         1
  Metadata Sequence No   4
  VG Access               read/write
  VG Status               resizable
  MAX LV                  0
  Cur LV                  3
  Open LV                 3
  Max PV                  0
  Cur PV                  1
  Act PV                  1
  VG Size                 20.00 GiB
  PE Size                 4.00 MiB
  Total PE                5119
  Alloc PE / Size        2860 / 11.17 GiB
  Free PE / Size         2259 / 8.82 GiB
  VG UUID                 x4Rht0-3b0s-6Zvm-CRtN-RwaS
```

Obr. 20. Volume group „source“

```
--- Logical volume ---
LV Name                  /dev/source/root
VG Name                  source
LV UUID                  NfX3U4-w5Vu-DSik
LV Write Access          read/write
LV Status                 available
# open                   1
LV Size                  952.00 MiB
Current LE                238
Segments                 1
Allocation                inherit
Read ahead sectors        auto
- currently set to       256
Block device              253:0
```

Obr. 21. LV ROOT

Kontrola, zdali je spuštěna služba *SSH*, lze provést příkazem `service ssh status` nebo příkazem `/etc/init.d/ssh status`, viz obr. 20. Konfigurace *SSH* serveru bude podrobněji popsána v části Návrh zabezpečení.

```
root@debian:~# service ssh status
sshd is running.
root@debian:~# /etc/init.d/ssh status
sshd is running.
root@debian:~# █
```

Obr. 22. *SSH* služba status

### 3 KONFIGURACE PROTOKOLŮ *NFS* A *SMB*

Síťové protokoly *NFS* a *SMB* byly v *Linuxových* a *Unixových* systémech vytvořeny pro sdílení adresářů a souborů. Následující kapitola se zabývá jejich podrobnějším popisem a popisem jejich konfigurace jak na serveru, tak na klientech.

#### 3.1 Protokol *NFS*

Protokol *NFS* slouží je sdílení souborů a aplikací v rámci počítačové sítě, jehož podstata je podobná systému sdílení disků ve *Windows NT*. Hlavní podobnost spočívá v tom, že oba operační systémy umožňují připojovat síťové disky a pracovat s nimi jako s disky lokálními.

Rozdíly sdílení souborů v *Linux* a *Windows NT* jsou hlavně ve způsobech správy souborů.

##### 3.1.1 Server a *NFS*

Vzhledem k tomu že v dnešní době většina dostupných distribucí *Linuxu* je dodávána přímo s nainstalovanou službou *NFS*, je dostačující provést přímo kontrolu, zdali je služba na daném serveru aktivní či nikoli.

Nejprve je důležité si ověřit, zda je služba na serveru spuštěna, k čemuž slouží příkaz `rpcinfo`. Pokud se zobrazí níže uvedený výpis, server *NFS* je spuštěn:

```

root@debian:~# rpcinfo -p
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 46358 status
100024 1 tcp 58146 status
100021 1 udp 45494 nlockmgr
100021 3 udp 45494 nlockmgr
100021 4 udp 45494 nlockmgr
100021 1 tcp 35099 nlockmgr
100021 3 tcp 35099 nlockmgr
100021 4 tcp 35099 nlockmgr
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
100003 4 tcp 2049 nfs
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 4 udp 2049 nfs
100005 1 udp 36359 mountd
100005 1 tcp 57712 mountd

```

Obr. 23. Server *NFS*

Pokud by server *NFS* nebyl spuštěn, chyběly by řádky *NFS* a *mountd*. Jestliže služba z jakéhokoliv důvodu neodpovídá a je aktivní, je možné provést její restart bez nutnosti restartu celého serveru a to příkazem `/etc/init.d/nfs-kernel-server restart`. Jestliže je z jakéhokoliv důvodu vyžadováno zastavení či spuštění serveru *NFS*, stačí zaměnit parametr `restart` za `stop` nebo `start`. [20]

### 3.1.2 Konfigurace *NFS*

*NFS* server je možné nastavit ve dvou krocích. Ten první je konfigurace souboru `/etc/exports`, který definuje, které části disku serveru budou sdíleny s ostatními uživateli sítě. Soubor také stanoví pravidla sdílení. V souboru `/etc/exports` je uvedeno, zda určitý uživatel má právo z disku číst nebo na něj i zapisovat. Druhým krokem je spuštění procesu serveru *NFS*, který se řídí instrukcemi zapsanými v souboru `/etc/exports`. [11]

#### 3.1.2.1 Konfigurace souboru `/etc/exports`

Server *NFS* má jediný konfigurační soubor a to `/etc/exports`. V tomto souboru je uveden seznam diskových oddílů, které lze sdílet, seznam uživatelů, kteří je smějí sdílet a jejich přístupová práva.

Formát položek v `/etc/exports` je následující:

**`/data/test client (permissions)`**

- **`/data/test`** nám určuje, který adresář je na serveru určen pro sdílení
- **`client`** jméno klienta, který je oprávněn ke sdílení, tedy je to jméno stanice, jež si připojuje daný adresář
- **`(permissions)`** v poslední části se nastavují práva, se kterými se uživatel může připojit

Parametr `(permissions)` může mít následující parametry:

- **`secure`** – číslo portu, po kterém klient požaduje připojení; musí být menší než 1024
- **`ro`** – přístupová práva uživatele do daného adresáře pouze pro čtení

- **rw** – přístupová práva uživatele do daného adresáře pro zápis
- **noaccess** – pokud zadáme /data/get/image klient se bude moci připojit do adresáře /data avšak adresář /data/get bude mít zakázán
- **no\_ROOT\_squash** – tento parametr se používá pro povolení přístupu superuživateli z klientské stanice k adresářům připojeným přes *NFS*
- **squash\_uids=uid-list** – tento parametr je použit v případě požadovaného omezení přístupu některých uživatelů, reprezentovaných jejich *UID* (*user identifier*), k určitým adresářům. Parametr může být zadán takto:  
squash\_uids=7, 9-11
- **squash\_gids=gid-list** – funguje stejně jako squash\_uids=uid-list, pouze s tím rozdílem, že používá ID skupin[11]

Tedy konfigurační soubor /etc/exports může v kompletní fázi vypadat takto:

```
# Example for NFSv4:
# /srv/nfs4      gss/krb5i(rw,sync,fsid=0,crossmnt,no
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/data/to/txt 192.168.154.133 (rw,squash_uids=1001)
/home/ivi/ 192.168.154.133 (rw,no_root_squash)
```

Obr. 24. Konfigurace /etc/exports

### 3.1.2.2 Načtení souboru /etc/exports

Pokud je soubor /etc/exports zeditovaný, je nutné dále provést export (načtení) jeho konfigurace. Pokud by nebyl tento krok proveden, tak by server *NFS* o těchto změnách nevěděl. Načtení se provede příkazem `exportfs`. Tento příkaz má samozřejmě i své parametry. Ty nejzákladnější jsou:

- **-a** – exportuje všechny záznamy v souboru /etc/exports
- **-u client:/data/get** – neexportuje uvedený parametr
- **-v** – zobrazí průběžné informace o procesu[11]

### 3.1.3 Konfigurace klienta NFS

Na klienta je možné na zvolené místo připojit sdílené adresáře serveru NFS, a to pomocí příkazu `mount`. Důležité však je, že tento příkaz může na klientské stanici spustit pouze superuživatel `ROOT`.

#### 3.1.3.1 Příkaz `mount`

Příkaz `mount` slouží pro připojování zařízení jako diskové oddíly, CD mechaniky, externí disky a jiné. Jeho syntaxe by se dala zapsat následovně:

```
mount <server:></co/></kam/>
```

**<server:>** zde se zadá název serveru, kterým se identifikuje v síti nebo je IP adresa

**</co/>** zadání adresáře, který je nadefinován na server NFS v konfiguračním souboru `/etc/exports`

**</kam/>** do kterého adresáře chceme sdílený oddíl připojit

Celý příkaz `mount` by měl tedy vypadat následovně:

```
mount 192.168.157.128:/data/to/txt /home/admin
```

Tím příkazujeme: připoj z NFS server 192.168.157.128 nesdílený adresář `/data/to/txt` do domovského adresáře uživatele `ADMIN`. Celý příkaz proběhne korektně pouze, pokud daný NFS klient má na NFS serveru příslušná práva. [11]

Kontrolu, zda `mount` připojil daný adresář z NFS serveru klientovi, je možné provést příkazem `df` s parametrem `-h`, viz obr. 23.

```
root@debian:/home/admin/get# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/source-root
                893M  582M  264M   69% /
tmpfs           252M    0  252M    0% /lib/init/rw
udev            247M  176K  247M    1% /dev
tmpfs           252M    0  252M    0% /dev/shm
/dev/mapper/source-data
                9.2G  150M   8.6G    2% /data
192.168.154.128:/data/to/txt
                4.6G  633M   3.8G   15% /home/admin
192.168.154.128:/home/ivi
                4.6G  633M   3.8G   15% /home/admin
```

Obr. 25. Příkaz `df -h`

Z obr. 23 je zřejmé, že příkaz `mount` proběhl korektně. Je vidět, že z *NFS* serveru byly připojeny dva oddíly:

- 1) `/data/to/txt` do adresáře `/home/admin`
- 2) `/home/ivi` taktéž do adresáře `/home/admin`

Vlastní instalací je ověřen názor autorů publikace *Administrace systému Linux* na protokol *NFS*. Tento protokol nepatří mezi nejbezpečnější, a to pro jeho omezené možnosti konfigurace. Z toho důvodu je nezbytné důvěřovat klientům ve společné síti a samozřejmě superuživateli klienta *NFS*. Pokud správce nemá důvěru k uživatelům, je nutné přistoupit k omezení práv a zpřístupnit soubory pouze pro čtení. Nicméně ne vždy to okolnosti dovolují.

## 3.2 Protokol SMB

Protokol *SMB* používají systémy na bázi *Windows*. V dnešní době je protokol *SMB* znám pod názvem *CIFS* (*Common Internet File System*). Pro zabezpečení komunikace mezi systémy *Windows* a *Linux* je potřebné nakonfigurovat server *SAMBA*. *SAMBA* totiž rozumí *Microsoft* protokolu *CIFS*. Pro správce to znamená, že můžeme nasadit *Linuxový* server, aniž by se na klienty systému *Windows* muselo instalovat *NFS*.

### 3.2.1 SAMBA server

*SAMBA* na *Linuxovém* serveru bývá součástí hlavní distribuce. Nastavení chování serveru se provádí konfigurací souboru `smb.conf`, který je v našem případě umístěn v `/etc/samba/smb.conf`. V tomto souboru se provádí celá konfigurace sdílení adresářů či tiskáren. Pokud správce *SAMBA* serveru potřebuje upravit práva do sdílených adresářů, nadefinovat nová práva nebo jakkoliv změnit pravidla sdílení adresářů a souborů, provede úpravu konfiguračního souboru `smb.conf`.

Pro správce *SAMBA* serveru je také nezbytné pro kontrolu jeho chování kontrolovat log soubor `log.smbd`, který je umístěn v `/var/log/samba/log.smbd`. Do tohoto souboru se zapisují všechny informace o chybách, spuštění či zastavení serveru. Správce zde najde podrobný výpis všech informací.

Démon `smbd` obsluhuje vlastní sdílení souborových systémů a tiskových služeb pro klienty. Nejprve se spojí s portem 139 a naslouchá požadavkům. Při každé autentizaci klienta se `smbd` zkopíruje, originál se vrací na port 139 pro nové požadavky a kopie obsluhuje připojení klienta. Tato nová kopie změní svůj aktuální identifikátor uživatele z uživatele `ROOT` na autentizovaného uživatele. Kopie zůstává v paměti tak dlouho, dokud trvá připojení klienta.

*SAMBA* kromě démona `smbd` obsahuje také `nmbd`. Tento démon je odpovědný za obsluhu požadavků jmenného serveru `NetBIOS`. Nejprve se spojí s portem 137 a na rozdíl od `smbd`, však nevytvoří pro každý dotaz instanci sama sebe.

Server *SAMBA* také umí pro svou komunikaci používat zašifrovaná i nezašifrovaná hesla. Je podstatné ověřit, jakou formou komunikují klienti využívající systémem *Windows* se serverem, používají-li šifrovanou či nešifrovanou formu. *Microsoft* začal používat zašifrovanou komunikaci teprve od doby uvedení distribuce *Windows NT 4.0/Service Pack 3*. Do této doby veškerá komunikace klientů byla nezašifrovaná, což neslo velká rizika. Nicméně i dnešní systémy jako *Windows 7* mohou po úpravě registrů využívat nezašifrovanou komunikaci. V této bakalářské práci bude pro větší bezpečnost sdílených dat přenášených po datové síti použita zašifrovaná komunikace.[16]

### 3.2.2 Konfigurace *SAMBA* serveru

Konfiguraci *SAMBA* serveru je prováděna editací souboru `smb.conf`, který se nachází v `/etc/samba/`, jak bylo uvedeno v předešlé části. Konfigurační soubor je členěn do čtyř sekcí, pod které se píšou samostatné konfigurační volby. Jsou to:

- 1) **[global]** – tato část určuje chování serveru *SAMBA* jako celku a provádí se zde konfigurace všech nabízených prostředků.
- 2) **[home]** – přikazuje každému *Linuxovému* uživateli připojit se do svého domovského adresáře.
- 3) **[název\_sdíleného\_prostředku]** – tato sekce vytváří samostatné sdílené adresáře, její název určuje název sdíleného prostředku.
- 4) **[printers]** – nastavení tiskáren.[11,21]

### 3.2.2.1 Konfigurace části [global]

Konfigurace části [global] je zvolena následující:

```
[global]
workgroup = WORKGROUP
server string = UTB-BT
netbios name = Debian
dns proxy = no
client code page = 852
character set = ISO8859-2
log file = /var/log/samba/log.%m
max log size = 1000
syslog = 0
SMB passwd file = /etc/smbpasswd
security = user
encrypt passwords = yes
```

Popis jednotlivých částí:

- 1) **workgroup** – název pracovní skupiny serveru v síti. Zde je ponecháno výchozí nastavení na `workgroup`. Všichni klienti musí být ve stejné skupině.
- 2) **server string** – komentář, který se objeví v okolních počítačích u jména serveru. Je na správci, jaký název zvolí pro komentář.
- 3) **netbios name** – jméno serveru, pod jakým bude v síti definován.
- 4) **DNS proxy** – tento parametr by se z výchozí hodnoty `no` měnil pouze v případě, že by byl použit oddělovač (*proxy server*) lokální sítě od Internetu.
- 5) **client code page** - tento parametr je nastaven na hodnotu 852, což je česká znaková sada a díky ní *SAMBA* podporuje češtinu ve sdílených adresářích.
- 6) **character set** – nastavením na hodnotu `ISO8859-2` je zajištěno, že se názvy souborů v systému serveru budou zobrazovat korektně, pokud v nich bude použita česká diakritika.
- 7) **log file** – logovací soubor *SAMBY* byl již zmíněn a pro snadnější orientaci na serveru je jeho umístění ponecháno ve výchozím nastavení tedy `/var/log/samba/log.%m`.
- 8) **max log size** – maximální velikost logovacího souboru v bajtech.

- 9) **syslog** - tento parametr nastavuje práh pro odesílání zpráv do systémového logovacího souboru *SAMBY*. Výchozí hodnota je 0, což znamená, že se budou ukládat všechny události.
- 10) **SMB passwd file** – cesta k souboru s hesly, které *SAMBA* bude používat. Výchozí hodnota v distribuci *Debian* je `/usr/bin/smbpasswd`, je zde provedena změna na `/etc/smbpasswd`, z důvodu jednoho umístění se souborem konfigurace systémových uživatelů.
- 11) **security** – jedná se o nejdůležitější hodnotu z hlediska zabezpečení. Zde je zvolen parametr `user`, což umožní připojení pouze existujících *Linuxových* uživatelů při použití sekce `[homes]` a také uživatelů nadefinovaných pod jednotlivými sdílenými prostředky pomocí `admin users` nebo `valid users`. Nikdo jiný nemá přístup a heslo je vázáno na konkrétního uživatele.
- 12) **encrypt passwords** – šifrování hesel, které je ve výchozím nastavení na hodnotě `no`, což znamená, že server bude při každé autentifikaci klienta předpokládat, že heslo není šifrované. Použití výchozí hodnoty se již dnes příliš nepoužívá vzhledem k tomu, že od verze *Windows NT 4.0/Service Pack 3* se již používá šifrování hesel.[16]

### 3.2.2.2 Konfigurace části `[homes]`

Nastavení této části slouží pro přidělení přístupu přes *SMB* do domovského adresáře na *Linuxovém* serveru. Každý nově založený uživatel na *Linuxovém* serveru má automaticky vytvořen svůj domovský adresář `/home/jméno_uživatele`.

**V práci je zvolena následující konfigurace:**

```
[homes]
comment = home
writetable = yes
browseable = no
valid users = %S
```

- 1) **comment** – tento parametr určuje, jaký se u adresáře bude zobrazovat komentář.

- 2) **writetable** – zde je určeno, jestli je možné do adresáře zapisovat či nikoliv. Vzhledem k tomu že se jedná o domovský adresář uživatele, tak je samozřejmě zvoleno povolení zápisu.
- 3) **browseable** – určuje, jestli bude při prohlížení sítě prostředek viditelný.
- 4) **valid users** – parametr %s nastaví klientovi jeho domovský adresář.[21]

### 3.2.2.3 Konfigurace části [sdílený adresář]

Tato část *SAMBY* je zcela v rukou správce. Podle požadavků zde administrátor definuje, které adresáře budou v rámci sítě sdílené a jaká na nich budou nastavena přístupová práva. Struktura této části konfiguračního souboru `smb.conf` je shodná s částí [homes]. Samozřejmě se liší v konfiguraci práv a definování sdíleného adresáře.[21]

Pro tuto práci je zvolena následující struktura. Je však na každém administrátorovi, jakou strukturu zvolí vzhledem k prostředí, kde se *SAMBA* implementuje a jaké jsou aktuální požadavky na sdílení.

```
[Accounting]
Comment = Accounting section
path = /data/Accounting
public = no
valid users = man_1, spicak
writetable = yes
browseable = yes
directory mask = 0700
create mask = 0700
```

```
[QC]
Comment = Quality control
path = /data/QC
public = no
valid user = man_2
write list = @managers
writetable = yes
```

```
browseable = yes
directory mask = 0770
create mask = 0770
```

```
[HRandGA]
Comment = HR and GA
path = /data/HRandGA
public = no
valid users = man_3
writetable = yes
browseable = yes
directory mask = 0700
create mask = 0700
```

```
[Public]
Comment = Public folder
path = /data/Public
public = yes
writetable = yes
browseable = yes
directory mask = 0700
create mask = 0700
```

### Nyní budou postupně popsány použité parametry:

**path** – tento parametr určuje cestu ke sdílenému adresáři.

**public** – hodnota `yes` definuje přístup všem uživatelům a hodnota `no` povolí přístup pouze oprávněným uživatelům.

**write list** – zde povolujeme přístup uživatelům, kteří jsou v určité skupině.

**directory mask** – povolení zápisu pro uživatele k adresáři pokud je hodnota `0700`, u hodnoty `0770` jsou práva zápisu povolena i pro skupinu.

**create mask** – uživatelem všechny nově vytvořené adresáře nebo soubory budou vytvořeny s právy zápisu pro uživatele, pokud je hodnota `0700`. U hodnoty `0770` jsou práva zápisu povolena i pro skupinu.

Parametry jako `comment`, `valid users`, `writetable` a `browseable` jsou popsány v předchozí části.[16]

### 3.2.3 Konfigurace uživatelů SAMBA serveru

Celé nastavení je směřováno tak, že klient, který se přihlásí ke sdíleným adresářům na SAMBA serveru, musí mít svůj účet přímo v systému a na SAMBA serveru. Tento postup je zvolen z důvodu bezpečnosti přístupů na datový server. Tedy každý klient se musí autentizovat pomocí uživatelského jména a hesla. Až na základě těchto požadavků mu je povolen přístup na server.

Vytvoření uživatele do systému *Linux* je proveden příkazem `usradd` uživatelské\_jméno. Pro výše uvedený příklad sdílených složek je příkazem `groupadd` založena skupina `managers`. Do zmíněné skupiny jsou vkládány účty manažerů jednotlivých oddělení. Celý příkaz `useradd` vypadá tedy takto:

```
useradd -c „Manager Accounting“ -g managers man_1
```

Pro kontrolu je zobrazen výpis souboru `/etc/passwd` pomocí příkazu `cat`. Na obr. 24 je zřejmé, že příkaz proběhl korektně. Byl vytvořen uživatel `man_1` i jeho domovský adresář `/home/man_1`, který tomuto uživateli bude automaticky připojen dle parametrů v `smb.conf` v části `[homes]`.

```
man_1:x:1002:1002:Manager Accounting:/home/man_1:/bin/sh
man_2:x:1003:1002:Manager QC:/home/man_2:/bin/sh
man_3:x:1004:1002:Manager HRandGA:/home/man_3:/bin/sh
man_4:x:1005:1002:Manager Maintenance:/home/man_4:/bin/sh
```

Obr. 26. Výpis `/etc/passwd`

Po úspěšném vytvoření uživatele v systému je nutné mu ještě povolit přístup k SAMBA serveru. To je provedeno příkazem `smbpasswd` s parametrem `-a`. Příkaz by měl vypadat následovně: `smbpasswd -a man_1`

Základní parametry příkazu `smbpasswd`:

- **-a** – přidání uživatele do souboru `/etc/smbpasswd`

- **-d** – zablokování účtu uživatele, a to pouze na *SAMBA* serveru nikoli v systému
- **-e** – odblokování účtu uživatele
- **-x** – odstranění uživatele ze *SAMBA* serveru
- **-n** – nastavení nového hesla

Samozřejmě parametrů je více, zde jsou uvedeny ty nejzákladnější a nejvíce používané.[16]

V tuto chvíli je *SAMBA* server připraven ke sdílení námi definovaných adresářů pro jednotlivé uživatele. Ve chvíli kdy administrátor má provedenou konečnou funkční konfiguraci *SAMBA* serveru, tak je vhodné provést zálohu konfiguračního `smb.conf` a uložit ji na bezpečné místo. Velmi vhodný postup pro uložení nastavení *SAMBA* serveru.

## 4 IMPLEMENTACE PODPORY *iSCSI*

Použití *iSCSI* je vhodné především u aplikací, které jsou schopné ukládat data pouze na fyzicky připojené disky, tedy nepodporují protokoly *SMB/CIFS* nebo *NFS*.

Norma *iSCSI* používá vlastní terminologii: Pokud u *SCSI* hovoříme o adaptéru a disku, adaptér nám nahradí komponent, který se jmenuje *Initiator* a cílové zařízení (např. disk/diskové pole, případně pásková jednotka), se nazývá cíl (*target*).

### 4.1 Instalace *iSCSI* a podpůrných modulů

Pro implementaci podpory *iSCSI* bylo tedy nutné do jádra operačního systému *DEBIAN* doinstalovat tři instalační balíčky pomocí příkazu:

```
apt-get install iscsitarget
apt-get install iscsitarget-source
apt-get install iscsitarget-dkms
```

Instalační balíček *iscsitarget-source* obsahuje zdrojový kód pro modul *iscsitarget* a *iscsitarget-dkms* a vytvoří knihovny pro *iscsitarget* přímo v jádru (*kernel*) systému.

```
make KERNELRELEASE=2.6.32-5-686 -C /lib/modules/2.6.32-5-686/build M=/var/lib/dkms/iscsitarget/1.4.20.2/build....._
```

Obr. 27. Instalace *iSCSI*target-dkms

Po úspěšné instalaci je důležité povolit službu editací souboru, který se nachází v `/etc/default/iscsitarget`. Tento soubor má jen jeden parametr `ISCSITARGET_ENABLE`, který je nastaven na hodnotu `FALSE`. Proto je tedy nutné tuto hodnotu změnit na `TRUE`, čímž je řečeno, že je služba `ISCSITARGET` povolena a bude automaticky spuštěna při každém restartu operačního systému.[9,22]

```
GNU nano 2.2.4      File: /etc//default/iscsitarget
ISC SITARGET_ENABLE=true
```

Obr. 28. *iSCSItarget enable*

Nyní je služba ISCSITARGET funkční a připravena ke konfiguraci sdílení. Na obr. 27 je zřejmé, že služba je funkční.

```
root@debian:~# /etc/init.d/iscsitarget restart
Removing iSCSI enterprise target devices: ::
Stopping iSCSI enterprise target service: ::
Removing iSCSI enterprise target modules: ... (warning).
.
Starting iSCSI enterprise target service:.
.
root@debian:~# /etc/init.d/iscsitarget status
iSCSI enterprise target is running.
```

Obr. 29. *iSCSI status*

## 4.2 Konfigurace /etc/iet/ietd.conf

Soubor /etc/iet/ietd.conf je hlavní konfigurační soubor služby ISCSITARGET, ve kterém jsou ve výchozím stavu všechny parametry skryty pomocí znaku #. Tento konfigurační soubor umožňuje nastavovat různé parametry, ať na úrovni celé služby tak i pro konkrétní *targety*.

### Konfigurace:

- **MaxConnections** – tento parametr určuje, kolik klientů může být přes *iSCSI* připojeno. Z důvodu bezpečnosti byl parametr nastaven na hodnotu 1. Pokud by v budoucnu došlo ke změně infrastruktury *iSCSI* (více klientů využívající propojení se serverem pomocí *iSCSI*), samozřejmě je možné tento parametr upravit na požadovanou hodnotu.
- **Target** – tato část nastavení je nejdůležitější částí konfiguračního souboru *ietd.conf*, protože se zde určuje, co se bude sdílet a za jakých podmínek. Důležité je to, že výchozí stav konfiguračního souboru nemá uvedeno žádné

sdílení, proto je nutné, aby vše vytvořil administrátor. Příklad nakonfigurované části *target*:

```
Target iqn.2013-4.com.data:storage.disk.iscsiLUN0
Incominguser spicak ivi
Lun 0 Path=/dev/skupina/iscsi_LUN0
Alias Lun 0
```

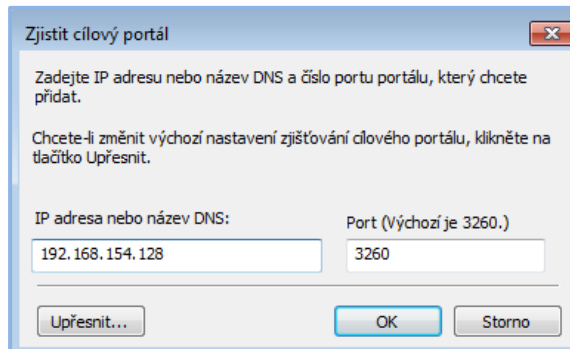
```
Target iqn.2013-4.com.data:storage.disk.iscsiLUN1
Lun 0 Path=/dev/skupina/iscsi_LUN1
Alias Lun 1
```

- **Iqn** – nám definuje název *targetu*, který je sdílen. Podle tohoto názvu je definována cesta ke sdílenému oddílu. Na straně klienta se tedy zadá název *targetu* a cíl na základě tohoto názvu nasdílí určený oddíl.
- **Incominguser** – syntaxe tohoto parametru je *<jméno><heslo>*, tedy klient se vůči cíli musí ověřit uživatelským jménem a heslem. Pokud klient není ověřen, nebude mu diskový prostor poskytnut.
- **Path** – tento parametr určuje, jaká část disku bude sdílena na základě přijatého jména *targetu*. Klient tedy zašle požadavek prostřednictvím jména *targetu* a server na základě tohoto jména určí, jakou část disku zpřístupní.[9, 22]

### 4.3 Připojení *iSCSI* ve *Windows*

Pro připojení sdíleného oddílu přes *iSCSI* byla zvolena distribuce *Windows 7*. *Windows* pro připojení používá *iSCSI initiator*. *iSCSI initiator* je utilita, která umožňuje systému *Windows* připojení k *iSCSI targetu*. Tedy k serveru, který zprostředkovává sdílení přes *iSCSI*. Pomocí následujících kroků se připojí *iSCSI target*:

- 1) Nejprve je nutné v *iSCSI initiatoru* zadat cílové připojení, tedy zadat *IP* serveru, který sdílí *iSCSI* targety. Port 3260 je výchozí port *iSCSI*.



Obr. 30. Cílový portál iSCSI

2) Po korektním spojení se serverem jsou zobrazeny *iSCSI targety*, které server sdílí.

Na obr. 31 je zřejmé, že byl zvolen *target*:

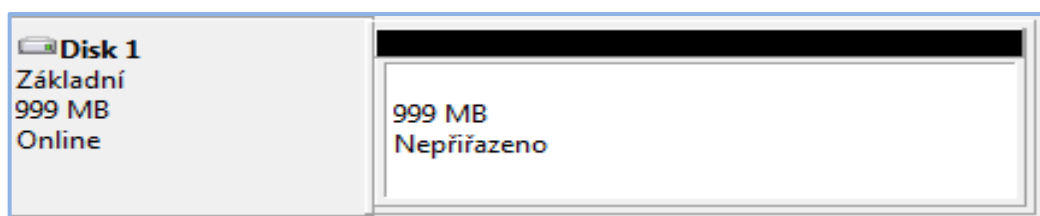
`iqn.2013-4.com.data:storage.disk.iscsiLUN1`.

Byl zvolen z důvodu, že nevyžaduje žádnou autentifikaci.

Název	Stav
<code>iqn.2013-4.com.data:storage.disk.iscsiLUN0</code>	Neaktivní
<code>iqn.2031-4.com.data:storage.disk.iscsiLUN1</code>	Připojeno

Obr. 31. Připojený target

V poslední části je nutné naformátovat připojený diskový prostor přes správce disků a přiřadit písmeno jednotky.



Obr. 32. Diskový prostor

Po aplikaci výše uvedených kroků je k dispozici nový disk, který se tváří jako by to byl disk lokální. A to je výhodou pro aplikace, které neumí komunikovat pomocí *SMB* nebo *NFS* protokolu. Nevýhodou může být, že je vyžadováno stabilní datové spojení.

## 4.4 Připojení *iSCSI* v *Linux Debian*

Nastavení připojení *iSCSI* cíle v *Linux* je složitější než jak tomu bylo ve *Windows 7*. Výhodou je podpora autentifikace, tedy větší bezpečnost při sdílení. V *Linuxu* je nutné doinstalovat modul `open-iscsi`. Instalace je provedena příkazem: `apt-get install open-iscsi`.

Po úspěšné instalaci `open-iscsi`, je umožněno přejít ke konfiguraci souboru `/etc/iSCSI/iscsi.conf`. Je tedy nezbytné nastavit tyto parametry:

- 1) `Node.startup` – tento parametr je ve výchozím nastavení skrytý. Je nutné ho povolit a nastavit na hodnotu `automatic`.
- 2) `Node.session.auth.authmethod` – je opět ve výchozím nastavení skrytý a je nastaven na hodnotu `CHAP`. Pro zabezpečení připojení *iSCSI* jednotky byl tedy tento parametr povolen. Zde je nutné nastavit uživatelské jméno a heslo pro připojení *targetu* v částech `discovery.sendtargets.auth.username` a `discovery.sendtargets.auth.password`.

Po nastavení výše uvedených parametrů se provede spuštění `open-iSCSI` příkazem `/etc/init.d/open-iscsi start.[9, 22]`

### 4.4.1 Kontrola připojení k serveru

Pokud je spuštěna služba `open-iscsi`, je nutné ověřit správnost nastavení `iscsid.conf`. Kontrola se provede zadáním příkazu `iscsiadm -m discovery -t st -p IP_ISCSI_SERVER`.

Z obr. 33 je zřejmé, že spojení se serverem bylo navázáno a že služba `iscsitarget` povolila klientovi přístup a následně zaslal výpis všech *targetů*.

```
root@debian:/etc/iscsi# iscsiadm -m discovery -t st -p 192.168.154.128
192.168.154.128:3260,1 iqn.2031-4.com.data:storage.disk.iscsiLUN1
192.168.154.128:3260,1 iqn.2013-4.com.data:storage.disk.iscsiLUN0
```

Obr. 33. Kontrolní výstup

#### 4.4.2 Připojení cíle *iSCSI* klientovi

Sdílený *target* se na klientův počítač připojí jako další fyzický disk, v tomto konkrétním případě jako `/dev/sdb` označený jako virtuální disk.

Nejprve je, však nutné připojit ke klientovi samotný *target* k čemuž použijeme příkaz:

```
iscsiadm --mode node --targetname iqn.2013-4.com.data:storage.disk
```

Na obr. 34 je vidět průběh připojení *iSCSI targetu*. Následně příkazem `lsscsi` je vypsán seznam použitých disků. Navíc je zde disk `/dev/sdb`. Posledním krokem před používáním virtuálního disku je jeho naformátování a připojení k některému adresáři. Pro provedení těchto kroků slouží následující příkazy:[9, 22]

```
mkfs.ext3 /dev/sdb
mount /dev/sdb /data
```

```
root@debian:~# iscsiadm -m discovery -t sendtargets -p 192.168.154.128
192.168.154.128:3260,1 iqn.2031-4.com.data:storage.disk.iscsiLUN1
192.168.154.128:3260,1 iqn.2013-4.com.data:storage.disk.iscsiLUN0
root@debian:~# iscsiadm --mode node --targetname iqn.2013-4.com.data:storage.disk.iscsiLUN0 --portal 192.168.154.128:3260 --login
Logging in to [iface: default, target: iqn.2013-4.com.data:storage.disk.iscsiLUN0, portal: 192.168.154.128,3260]
Login to [iface: default, target: iqn.2013-4.com.data:storage.disk.iscsiLUN0, portal: 192.168.154.128,3260]: successful
root@debian:~# lsscsi
[1:0:0:0]    cd/dvd  NECVMWar  VMware IDE CDR10  1.00  /dev/sr0
[2:0:0:0]    disk    VMware,  VMware Virtual S  1.0   /dev/sda
[3:0:0:0]    disk    IET       VIRTUAL-DISK      0     /dev/sdb
```

Obr. 34. Virtuální disk

Nyní klient může plně využívat diskový prostor, který mu prostřednictvím *iSCSI* byl poskytnut. Důležité je také to, že data na serverech jsou fyzicky oddělená a pravidelně zálohovaná. Pokud by došlo k poškození systému klienta, tak klient neztratí data uložená na virtuálním disku. V dnešní době je tento způsob připojení disků vyžíván i různými zálohovacími systémy, kterým je například *veeam backup and replication*.

## 5 NÁVRH ZABEZPEČENÍ

Tato část bakalářské práce je zaměřena na zabezpečení *Linuxového* serveru proti napadení zvenčí, omezení jeho provozu či zneužití jeho datové struktury.

Následující kapitola je soustředěna především na konfiguraci *SSH (Secure Shell)* serveru, který je využíván pro vzdálenou správu, proto prostřednictvím této komunikace může dojít k odhalení hesel.

Dalším krokem je možnost blokace klientů, kteří se z jakéhokoliv důvodu několikrát chybně přihlásí třeba i přes webové rozhraní prostřednictvím protokolů (*FTP, HTTP, HTTPS*) nebo *SSH*.

### 5.1 Bezpečnost *SSH* serveru

*SSH* je v informatice označení pro program a současně pro zabezpečený komunikační protokol v počítačových sítích, které používají *TCP/IP*. *SSH* byl navržen jako náhrada za *telnet* a další nezabezpečené vzdálené *shelly (rlogin, rsh* apod.), které posílají heslo v nezabezpečené formě a umožňují tak jeho odposlechnutí při přenosu pomocí počítačové sítě. Šifrování přenášených dat, které *SSH* poskytuje, slouží k zabezpečení dat při přenosu přes nedůvěryhodnou síť, jako je například Internet.

*SSH* umožňuje bezpečnou komunikaci mezi dvěma počítači, která se využívá pro zprostředkování přístupu k příkazovému řádku, kopírování souborů a též jakýkoliv obecný přenos dat (s využitím síťového tunelování). *SSH* zabezpečuje autentizaci obou účastníků komunikace, transparentní šifrování přenášených dat, zajištění jejich integrity a volitelnou bezztrátovou kompresi. Server standardně naslouchá na portu *TCP/22*. [23]

#### 5.1.1 Návrh zabezpečení *SSH*

Nastavení *SSH* serveru se provádí konfigurací souboru *sshd\_config*, který je umístěn v */etc/ssh/sshd\_config*. Pro případ této práce je zamezení přístupu na server realizováno prostřednictvím *SSH* jako superuživatel *ROOT*. Jak již bylo zmíněno *SSH* naslouchá na portu 22. Tento port může být odposloucháván, pokud dojde k přihlášení jako superuživatel *ROOT*, může být heslo dešifrováno a následně zneužito

k napadení serveru. Z důvodu uvedeného nebezpečí je provedena změna konfigurace souboru `sshd_config` tak, aby byl odepřen superuživateli *ROOT* přístup, pokud se chce přihlásit přes *SSH*.

#### Změna je provedena v části:

```
# Authentication
PermitROOT Login no
```

Ve výchozím stavu je parametr `PermitROOT Login no` nastaven na hodnotu `yes`. Po této změně je nutné *SSH* server restartovat tak, aby se změny projevíly okamžitě. Restart *SSH* server je proveden příkazem `/etc/init.d/ssh restart`.

Na obr. 35 lze vidět, že při pokusu o přihlášení na server prostřednictvím *SSH* jako superuživatel *ROOT*, byl jeho přístup zamítnut. Přesně řečeno nemá práva pro přihlášení.

```
login as: root
root@192.168.154.128's password:
Access denied
root@192.168.154.128's password: █
```

Obr. 35. Nepovolený přístup

Pokud je tedy vyžadováno přihlášení na server prostřednictvím *SSH*, je nutné použít jiný uživatelský účet. Ve zmiňovaném případě je použit již vytvořený účet uživatele *UTB*. Obr. 36 potvrzuje, že uživateli *UTB* byl přístup prostřednictvím *SSH* povolen.

```
login as: utb
utb@192.168.154.128's password:
Linux debian 2.6.32-5-686 #1 SMP Mon Feb 25 01:04:36 UTC 2013 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

Obr. 36. Přihlášení k serveru pře *SSH* jako uživatel *UTB*

Nyní pokud je nutné z důvodu nastavení systému nebo konfigurace na serveru pracovat jako superuživatel *ROOT*, tak pomocí příkazu `su` je možné se pouze přepnout.

Tak jak je zřejmé na obr. 37, po zadání příkaz je systémem požadováno heslo na superuživatele.

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr 17 23:20:14 2013 from 192.168.154.1
utb@debian:~$ su
Password:
root@debian:/home/utb#
```

Obr. 37. Příkaz *su*

Předešlými kroky bylo zamezeno možnosti dešifrování hesla superuživatele odposloucháváním na portu 22. Nyní pokud by bylo dešifrováno heslo uživatele UTB tak, již není tak vysoké nebezpečí při pokusu o napadení serveru. Uživatel nemá taková práva, která by mu umožnila jakkoliv ovlivnit či omezit chod serveru.

Možností zabezpečení *SSH* je samozřejmě více, nicméně nevhodnější je kombinace různých druhů zabezpečení. Další možnost bude popsána v následující části.

## 5.2 Implementace služby *FAIL2BAN*

*Daemon Fail2ban* slouží k blokadě *IP* adres snažících se o neoprávněný přístup k systému. *Fail2ban* běží na pozadí jako *daemon*, kdy kontroluje systémové logy a vyhodnocuje podle zadaných pravidel, zda se jedná o útok na server či nikoliv. Tento *daemon* dokáže ochránit nejen *SSH*, ale i řadu dalších služeb jako jsou například *FTP*.

### 5.2.1 Instalace a konfigurace *Fail2ban*

Instalaci *daemon* a *Fail2ban* provedeme příkazem `apt-get install fail2ban`. Po spuštění příkazu se systém připojí na *FTP* server, odkud stáhne požadovaný instalační balík. Po úspěšné instalaci je možné přejít k jeho konfiguraci.

V *Debianu* se konfigurace nástroje *Fail2ban* nachází v adresáři `/etc/fail2ban` a nastavení se provádí změnou konfigurace v souboru `jailed.conf`. Vzhledem k tomu, že

primárně na tomto serveru bude využíváno *SSH* pro vzdálenou správu, změny v konfiguračním souboru `fail2ban.conf` budou provedeny primárně na tuto službu. Dále budou popsány některé části konfiguračního souboru:

- [DEFAULT] – v této části je asi nedůležitější položka `ignoreip`, která je standardně nastavena na *IP* 127.0.0.1, což je *IP localhost*. V tomto případě tedy nikdy nedojde k zablokování této *IP*. Pokud je třeba zvolit ještě další *IP* adresu, u které není vyžadováno, aby ji *Fail2ban* blokoval, může být zapsána za *IP* 127.0.0.1. Je nutné *IP* adresy od sebe oddělit mezerou.

**Bantime = 600** – tento parametr určuje, na jakou dobu budou *IP* adresy blokovány.

- [SSH] – v této části konfiguračního souboru se nastavují pravidla pro *SSH*. Zde budou popsány jednotlivé parametry:

**Enableb = true** – tento parametr povoluje kontrolu *SSH*, pokud by bylo nutné tuto kontrolu zakázat, stačí parametr `true` změnit na `false`. Nicméně tento krok se nedoporučuje.

**port = SSH** - zde bude pro danou *IP* adresu zablokován pouze *port* 22, ale pokud by parametr `SSH` byl změněn na `all`, tak daná *IP* bude mít blokován všechny porty.

**filter = SSH d** – tato volba označuje filtr, který se má u dané služby použít, což v tomto případě je na klienty.

**logpath = /var/log/auht.log** – zde se specifikuje umístění souboru s příslušným logem.

**Maxretry = 3** – tento parametr určuje počet pokusů pro přihlášení. Pokud uživatel čtyřikrát chybně zadá své heslo, *Fail2ban* ho zablokuje na 10 minut tak, jak je nastaveno v části [DEFAULT]. [24]

```
2013-04-18 19:12:16,315 fail2ban.jail : INFO Jail 'ssh' started
2013-04-18 19:23:09,864 fail2ban.actions: WARNING [ssh] Ban 192.168.154.1
root@debian:~# _
```

Obr. 38. *Fail2Ban.log*

V logovacím souboru `fail2ban.log` je důkaz toho, že nastavení *Fail2ban* pro *SSH* je funkční. Výpis *logu* říká, že *IP* 192.168.154.1 je blokována na základě chybného

přihlášení pře *SSH* . Pokud by server byl zároveň i emailovým serverem, je možné také nastavit odesílání zpráv o blokováních *IP* na určený email pomocí parametru `sendemail`. Tento parametr je možné nastavit na všechny části konfiguračního souboru nebo jen na některé služby. V případě této práce to může být část `[SSH]`. Nastavení parametru `sendemail` by mohlo vypadat takto: `sendemail-whois[name=SSH, dest= admin@utb.cz]`.

## 6 POWER MANAGEMENT

Otázka power managementu je v dnešní době hodně diskutovaná z hlediska nákladů na energie. Konfigurace power managementu je v oblasti výpočetní techniky využívána především z pohledu snižování nákladů. Pokud hardware umožňuje určitá nastavení, tak je schopen přejít do stavu s nízkou spotřebou, pokud se některé prostředky v danou chvíli nevyužívají na 100 %. Jedná se většinou o disková zařízení nebo procesory.

Pokud je snížena taktovací frekvenci procesoru, tak se sníží teplota procesoru a tím pádem chlazení celého serveru nemusí běžet na plný výkon. Samozřejmě vše se musí nastavit tak, aby nebyl nijak omezen celkový výkon serveru. Jestliže tedy jakákoliv aplikace v danou chvíli potřebuje maximální početní výkon procesoru, tak by ho měl poskytnout.

Následující tři podkapitoly se zaměřují na problematiku uspávání disků, změna taktovací frekvence procesoru a regulace otáček větráků.

### 6.1 Uspávání disků

Pro konfiguraci uspávání pevného disku slouží systémový příkaz `hdparm`. Stačí jej zavolat s parametrem `-s` a údajem o čase. Ten je vyjádřen číslem 0 – 255. Vyjadřuje časové úseky a je odstupňován v několika úrovních:

- **0** – pokud je použita tato hodnota, tak se disk nezastaví
- **1-240** – jedna jednotka je 5 vteřin, pokud je zadána hodnota 10, tak disk zastaví za 50 vteřin
- **241-251** – v tomto případě je jednotka 30 minut
- **252** – hodnota je rovna 21 minutám

Samozřejmě je nutno uvést, že uspávání disků se spustí pouze, pokud na daný disk nepřichází žádný požadavek pro čtení nebo zápis. Nicméně uspávání nemá žádný vliv na dostupnost zařízení sdíleného pomocí *iSCSI*. Pokud se disk uspí a klient zašle požadavek na zápis nebo čtení, reakce disků je okamžitá, tudíž uživatel nezaznamená žádné omezení dostupnosti.[25]

Nejprve je nutné nainstalovat nástroj `hdparm` (`harddisc parameter`), který umožňuje nastavení parametrů pevného disku. Instalace je provedena příkazem `apt-get install hdparm`. Nyní je vhodné zobrazit podrobné informace o pevných discích,

keré je možné zobrazit pomocí příkazu `hdparm -i /dev/sda1` a `hdparm -i /dev/sdb1`. [25]

```

root@debian:~# hdparm -i /dev/sdb1

/dev/sdb1:

Model=SAMSUNG SP1614C, FwRev=SW100-27, SerialNo=0696J1FX918280
Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs }
RawCHS=16383/16/63, TrkSize=34902, SectSize=554, ECCbytes=4
BuffType=DualPortCache, BuffSize=8192kB, MaxMultSect=16, MultSect=16
CurCHS=16383/16/63, CurSects=16514064, LBA=yes, LBAsects=312579695
IORDY=on/off, tPIO={min:120,w/IORDY:120}, tDMA={min:120,rec:120}
PIO modes:  pio0 pio1 pio2 pio3 pio4
DMA modes:  mdma0 mdma1 mdma2
UDMA modes: udma0 udma1 udma2 udma3 udma4 udma5 *udma6
AdvancedPM=no WriteCache=enabled
Drive conforms to: ATA/ATAPI-7 T13 1532D revision 0:  ATA/ATAPI-1,2,3,4,5,6,7

* signifies the current active mode

```

Obr. 39. Informace o HDD

### Pro nastavení uspávání disků je nutné spustit tyto příkazy:

```
hdparm -S 12 /dev/sda1
```

```
hdparm -S 12 /dev/sdb1
```

Výše uvedené příkazy nastaví uspání disků po jedné minutě, za předpokladu, že na ně za tuto dobu nepříjde žádný požadavek. Stanovení hodnoty je individuální a na zvaženi každého administrátora systému.

Vzniká zde ovšem jeden zásadní problém. Je nutné nastavit příkazy tak, aby pokud dojde k restartu systému, byly spuštěny automaticky. Proto, aby správce nemusel na nastavení uspávání myslet, je nutné vytvořit startovací skript.

V souboru `/etc/init.d/README` je návod jak by měl soubor se skriptem vypadat a kde má být umístěn. Pomocí příkazu `nano /etc/init.d/hdparm.sh` a je do něj zapsána následující syntaxe:

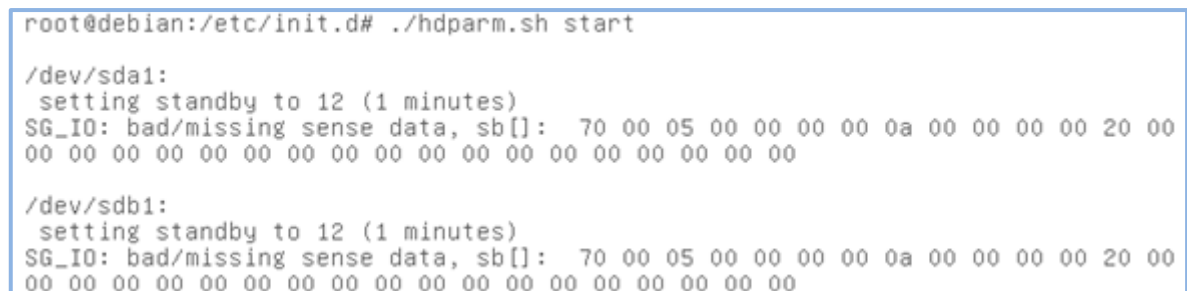
```

#! /bin/sh
### BEGIN INIT INFO
# Provides:          Jmeno
# Default-Start:    3 4 5
# Short-Description: hdparm script

```

```
# Description:          Dlouhy popis
### END INIT INFO
case "$1" in
start)
hdparm -S 12 /dev/sda1
hdparm -S 12 /dev/sdb1
;;
*)
echo  "Usage:  /etc/init.d/atd  {start|stop|restart|force-
reload|status}"
exit 1
;;
esac
exit 0
```

Nyní je možné tímto příkazem `./hdparm.sh start` ověřit, zda je popisovaný soubor funkční. Jestliže se zobrazí výpis postupného spouštění příkazů zapsaných v `hdparm.sh` jak je uvedeno na obr. 40, tak skript proběhl korektně.[25]



```
root@debian:/etc/init.d# ./hdparm.sh start
/dev/sda1:
  setting standby to 12 (1 minutes)
SG_IO: bad/missing sense data, sb[]:  70 00 05 00 00 00 00 00 0a 00 00 00 00 20 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
/dev/sdb1:
  setting standby to 12 (1 minutes)
SG_IO: bad/missing sense data, sb[]:  70 00 05 00 00 00 00 00 0a 00 00 00 00 20 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Obr. 40. Kontrola `hdparm.sh`

Aby se `hdparm.sh` spustil automaticky při startu systému, je nutné provést odkazy na tento soubor pomocí příkazu `update-rc.d hdparm.sh start`. Nyní již administrátor po restartu serveru nemusí myslet na ruční spouštění příkazu `hdparm`. Vše proběhne automaticky při každém spuštění systému.[25]

## 6.2 Změna taktovací frekvence procesoru

Změnou taktovací frekvence procesoru je možné docílit nejen nižší spotřeby elektrické energie, ale také snížení hlučnosti serveru. Pokud se sníží frekvence, zároveň se sníží teplota procesoru a tedy i nároky na chlazení jsou nižší. Tedy automaticky se sníží výkon chlazení, což zároveň sníží i hlučnost.

### 6.2.1 Konfigurace

Pro možnost změny chování procesoru je nutné do *Linuxové* distribuce *Debian* doinstalovat následující sadu nástrojů: `apt-get install CPUfrequtils sysfsutils`.

Nyní je nutné zvolit správný ovladač k danému procesoru, pokud systém nepřidělí ovladač automaticky, informace o *CPU* se vypíše pomocí zadání příkazu `cat /proc/cpuinfo`. Tento příkaz vypíše veškeré informace o procesoru. Pokud ovšem je vyžadován výpis jen některých informací, tak příkaz je nutné zadat následovně `cat /proc/cpuinfo | grep "model name"`. Nyní je vypsán pouze model procesoru ze souboru `/proc/cpuinfo`.

Následně je vhodné si zobrazit podrobné informace o procesoru ve smyslu zjištění použitého ovladače nebo možnosti nastavení frekvence. Použitý procesor Intel Atom CPU D525 je schopen pracovat na frekvencích 225 MHz, 450 MHz, 675 MHz, 900 MHz, 1.13 GHz, 1.35 GHz, 1.58 GHz, 1.80 GHz. Tyto a spoustu dalších informací nám zobrazí příkaz `cpufreq-info`.

Nastavení power managementu procesoru je realizována editací konfiguračního souboru `sysfs.conf`, který je umístěn v adresáři `/etc/`. Do tohoto konfiguračního souboru dopíšeme následující řádky:

```
devices/system/cpu/cpu0/cpufreq/scaling_governor = userspace
devices/system/cpu/cpu0/cpufreq/scaling_setspeed = 900000
devices/system/cpu/cpu1/cpufreq/scaling_governor = userspace
devices/system/cpu/cpu1/cpufreq/scaling_setspeed = 900000
devices/system/cpu/cpu2/cpufreq/scaling_governor = userspace
devices/system/cpu/cpu3/cpufreq/scaling_governor = userspace
```

- **devices/system/cpu/cpu\*** - CPU0 až CPU3 nám představuje jednotlivá jádra procesoru.
- **Scaling\_governor** – volně přeloženo z angličtiny jako měnič frekvence. Je to proměnná, která se používá s hodnotou `ondemand`, `conservative`, `userspace`, `performance`, `power save` nebo `interactive`.  
Ve výchozím stavu procesory pracují v režimu `performance`. Což znamená, že procesor nepřetržitě pracuje na nejvyšší frekvenci. V případě této práce byla zvolena hodnota `userspace`, která frekvenci procesoru nastaví na námi zvolenou hodnotu.
- **Scaling\_setspeed** – pomocí této proměnné je nastavena statická frekvence procesoru. Byla zvolena střední hodnota frekvence, což je 900 MHz.[26, 27]

Výše zmíněné nastavení je možné ověřit příkazy:

```
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
cat /proc/cpuinfo | grep MHz
```

```
root@debian:~# cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
userspace
userspace
userspace
userspace
root@debian:~# cat /proc/cpuinfo | grep MHz
cpu MHz          : 900.000
cpu MHz          : 900.000
cpu MHz          : 900.000
cpu MHz          : 900.000
```

Obr. 41. Kontrolní výpis

Zmíněný postup zajistí částečné snížení spotřeby elektrické energie a hlučnosti serveru. Nicméně výběr volby nastavení procesoru je závislý především na jeho hardwarové konfiguraci. Tedy na tom, v jakých režimech umí pracovat, což definuje přímo výrobce daného procesoru.

## 6.3 Regulace otáček větráku

Regulace otáček větráku je závěrečnou podkapitolou tématu konfigurace power managementu. Je zaměřena na konfiguraci systému tak, aby byl schopný ovládat prvky chlazení serverů a funkce plně využít. Samozřejmě je zde také otázka úspory energií, která je velmi důležitá a zásadní. Tedy pokud snížíme otáčky chladiče, tak tím snížíme nároky na energie a také snížíme hlučnost celého serveru.

### 6.3.1 Instalace lm-sensors

Jako první je nutné nainstalovat instalační balíček `lm-sensors`, který nám umožní přístup k informacím o teplotách, napětí a otáčkách větráků, jimiž server disponuje. Výše zmíněný instalační balíček se nainstaluje pomocí příkazu:

```
apt-get install lm-sensors
```

Po úspěšné instalaci je nutné provést detekci dostupného hardwaru. Detekce je realizována spuštěním příkazu `sensors-detect`, veškerá načtená data se uloží do konfiguračního souboru `/etc/sensors.conf`. Po načtení hardwarové konfigurace systém vyzve ke spuštění `/etc/init.d/module-init-tools start`

Pokud je modul korektně spuštěn, je důležité načíst nadetekovanou konfiguraci, a to příkazem `sensors -s`. Ověření kompletní funkčnosti proběhne po spuštění příkazem `sensors`. Tento příkaz vypíše veškeré informace o použitém hardwaru ve formě teplot, napětí a otáček chladičů jak je uvedeno na obr. 42.[28, 29]

```
fan1:          0 RPM (min = 0 RPM)
fan2:         12980 RPM (min = 0 RPM)
temp1:        -55.0Å°C (low = +127.0Å°C, high = +127.0Å°C) sensor = thermistor
temp2:         +64.0Å°C (low = +127.0Å°C, high = +127.0Å°C) sensor = thermistor
temp3:         +52.0Å°C (low = +127.0Å°C, high = +127.0Å°C) sensor = thermal di
```

Obr. 42. Příkaz `sensors`

### 6.3.2 Konfigurace rychlosti otáček chlazení

Pro vlastní konfiguraci otáček chlazení je použit příkaz `pwmconfig` a nastavení je nutné provést dle instrukcí. Veškeré použité nastavení je možné ověřit v konfiguračním souboru `/etc/fancontrol`, v němž je možné provádět změny nastavení. Nicméně je vhodnější jej používat spíše pro doladění potřebné konfigurace.

Na obr. 43 je výpis z konfiguračního souboru `/etc/fancontrol`. Parametry použité v souboru jsou následující:

```
INTERVAL=10
DEVPATH=hwmon2=devices/platform/it87.656
DEVNAME=hwmon2=it8720
FCTEMPS= hwmon2/device/pwm1=hwmon2/device/temp2_input
FCFANS= hwmon2/device/pwm1=hwmon2/device/fan1_input
MINTEMP= hwmon2/device/pwm1=70
MAXTEMP= hwmon2/device/pwm1=85
MINSTART= hwmon2/device/pwm1=0
MINSTOP= hwmon2/device/pwm1=0
MINPWM=hwmon2/device1=0
MAXPWM=hwmon2/device/pwm1=255
```

Obr. 43. Konfigurace `/etc/fancontrol`

- `INTERVAL` – tato volba označuje dobu mezi sledováním teplot a je v sekundách.
- `FCTEMPS` – zde jsou uvedeny použité teplotní senzory.
- `FCFANS` – kontrola otáček použitého větráku.
- `MINTEMP/MAXTEMP` – jedny z nejdůležitějších parametrů, prostřednictvím kterých se určuje rozsah teplotních limitů, jejich prostřednictvím se budou řídit otáčky větráku.
- `MINSTART` – nastaví minimální rychlost, při které se větrák začíná otáčet. Je vhodné použít bezpečnou hodnotu tak, aby bylo jisté, že i když dojde k opotřebení komponent větráku bude zajištěna jeho funkčnost.
- `MINSTOP` – minimální rychlost, při které se ještě větrák otáčí.
- `MINPWM/MAXPWM` – pomocí těchto parametrů se určí rozsah otáček větráku.[28, 29]

Nyní po nastavení, pokud není možné server restartovat, je nezbytné kontrolu větráku spustit ručně pomocí příkazu `fancontrol`, jehož výpis můžeme vidět na obr. 44.

```
Common settings:
  INTERVAL=10

Settings for hwmon2/device/pwm1:
  Depends on hwmon2/device/temp2_input
  Controls hwmon2/device/fan1_input
  MINTEMP=70
  MAXTEMP=85
  MINSTART=0
  MINSTOP=0
  MINPWM=0
  MAXPWM=255
```

Obr. 44. Výpis příkazu `fancontrol`

Pokud dojde k restartu serveru, kontrola větráku se již spustí automaticky načtením konfiguračního souboru `/etc/fancontrol`. Z uvedeného nastavení je zřejmé, že pokud teplota na senzoru `temp2_input` bude pod hranicí 70 °C, tak větrák `fan1_input` bude mít nulové otáčky. Naopak větrák bude mít maximální otáčky, pokud teplota na `temp2_input` bude rovna nebo přesáhne hodnotu uvedenou v parametru `MAXTEMP`.

Závěrem je nutné poznamenat, že otáčky větráku jsou závislé na námi nastaveném teplotním rozsahu zvoleného senzoru `temp2_input`. [28, 29]

## ZÁVĚR

Instalace *Linuxového* serveru byla fundamentálním prvkem celé bakalářské práce. Bylo zásadní se rozhodnout, jakou distribuci operačního systému *Linux* použít čili, která distribuce bude nejvhodnější pro konfiguraci *NAS*. Již před instalací bylo rozhodnuto, že bude zvolena právě distribuce *Debian*. Oproti distribuci *Fedora* má *Debian* mnohem propracovanější část instalace, co se týče vytvoření *RAIDu* a konfigurace *Logical Volume Management*.

První problém při instalaci *Linuxu* byl dán nepřítomností *CD* mechaniky v šasi serveru. Toto šasi nebylo uzpůsobeno pro připojení interní mechaniky, z toho důvodu bylo nezbytné navrhnout jiný způsob instalace. Jednou z možností bylo využít externí mechaniku, ale byla vybrána odlišná alternativa, která spočívala ve vytvoření *bootovacího* USB flash disku. Zmíněný *bootovací* USB flash disk byl vytvořen pomocí programu *UNetbootin*. Ačkoliv existuje celá řada podobných programů, *UNetbootin* byl zvolen především proto, že není zapotřebí jej instalovat a po jeho stažení je tedy možné jej okamžitě používat.

Při následné instalaci bylo potřeba se rozhodnout, jakým směrem se bude postupovat při rozdělování disku a zvolení vhodného typu *RAIDu*. Vzhledem k tomu, že server byl osazen dvojicí *SATA* disků, tak byly možné jen dvě varianty, a to *RAID 0* nebo *RAID 1*. Byl zvolen *RAID 1*, kdy sice nebyla nevyužita celá kapacita obou disků, nicméně prioritou bylo zamezit zastavení serveru při výpadku jednoho z disků. Jinými slovy primární byla bezpečnost dat a zabezpečení stability, což je v případě serveru velmi důležité a zásadní.

Poté byla v části rozdělení disků zvolena cesta vytvoření *volume group* přes celý *RAID 1*. Pro další rozšiřování diskové kapacity serveru je to nejvhodnější způsob a ze zkušeností může být potvrzeno, že v praxi je nejvíce využíván. Kdyby k použitému serveru byly přidány další dva disky, což je technicky možné, protože šasi je vybaveno čtyřmi sloty pro pevné disky, bylo by velmi jednoduché rozšířit diskovou kapacitu již vytvořené *volume group*. Pro správu a rozšiřování kapacity je to jedna z nejvhodnějších možností.

Navazující část práce byla zaměřena na konfiguraci protokolu *NFS* a *SMB*. U protokolu *SMB* je částečně složitější konfigurace, nicméně počáteční problémy s konfigurací se správci zúročí následnou bezpečností tohoto protokolu. U *NFS* je úroveň

zabezpečení podstatně nižší, avšak konfigurace tohoto protokolu probíhá nekomplikovaně. Proto se doporučuje, aby server určený pro sdílení dat ve větších společnostech, tato data sdílel prostřednictvím protokolu *SMB*. Nastavení přístupů pro různé skupiny uživatelů a samotné uživatele je v tomto protokolu velmi propracované.

Dalším krokem byla implementace podpory *iSCSI*. Síťový protokol *iSCSI* není příliš vhodný při sdílení dat mezi více klienty, a to především z toho důvodu, že je složitější konfigurace na straně klientů a na straně serveru je minimální možnost zabezpečení. Pro sdílení dat více uživatelům je vhodnější protokol *SMB*. Nicméně protokol *iSCSI* je vyhovující pro použití v rámci virtualizace serveru, kdy celá hierarchie *VMware* je většinou postavena na *iSCSI*. Využití *iSCSI* je možné v situaci, kdy jsou vytvořeny dvě virtuální sítě, síť externí a interní. Vhodné je použít *iSCSI* v interní síti pro propojení ESX serveru a sdílených datových úložišť. Dále je možné jeho využití pro zálohování virtuálních serverů prostřednictvím *Veeam Backup and Replication*. *iSCSI* se také používá v případech, kdy není možné využít komunikaci prostřednictvím *Fibre Channel*, buď z finanční, či technické nedostatečnosti.

Při hodnocení návrhu zabezpečení bylo nezbytné zvážit prostředí, ve kterém se bude server používat. Jestli server bude přístupný pouze v rámci interní sítě, nebo zda bude server sdílet data prostřednictvím internetu. Nejnebezpečnější je sdílení dat prostřednictvím internetu, kdy může dojít k pokusům server nějakým způsobem vyřadit z provozu např. prostřednictvím *DoS* útoku. Především proto byl zvolen zákaz přihlášení uživatele *ROOT* prostřednictvím *SSH* a konfiguraci *Fail2ban* pro blokaci *IP* adres po chybném přihlášení na server.

V otázce volby adekvátního typu nastavení power managementu je nutné poznamenat, že jakákoliv snaha o konfiguraci power managementu musí být podporována použitým hardwarem. Což se potvrdilo i v případě této práce při pokusu konfigurace škálování frekvence procesoru. Zde použitý procesor Intel Atom D525 tuto funkci nepodporoval, proto bylo ve snaze o snížení elektrické náročnosti serveru přistoupeno k pevnému snížení frekvence procesoru - tak zvanému „podtaktování“. Vzhledem k tomu, že celá bakalářská práce se zabývá konfigurací *NAS*, kde se předpokládá sdílení dat prostřednictvím počítačové sítě, není pevné nastavení frekvence procesoru považováno za zásadní problém. Tato volba by nebyla vhodná, pokud by na serveru byly spuštěny aplikace, které potřebují plnou početní kapacitu procesoru, i když jen na pár minut denně. Pak by bylo vhodné použít škálování frekvence procesoru, kdy se frekvence mění dle

nároků jednotlivých aplikací samozřejmě s použitím jiného procesoru, který tuto funkcionalitu podporuje.

Kromě „podtaktování“ procesoru byla zvolena i možnost nastavení otáček větráku. Při nastavování otáček bylo vhodné zvolit teplotní senzor procesoru, kde dochází k nejčastějším změnám teplot. Je vhodné upravit otáčky jen některých ventilátorů, protože pokud by došlo k závadě na teplotním senzoru a ten by nereagoval na změnu teplot, mohlo by dojít k přehřátí a následnému poškození některých vnitřních hardwarových komponent serveru. V průběhu konfigurace byl odhalen problém, že ovladač operačního systému umožňuje ovládání pouze jednoho větráku a to *CPU FAN*.

V době ukončení v této bakalářské práci byla vydána nová verze operačního systému *Linux Debian 7.0*. V návaznosti na řešení problematiky power managementu proběhlo testování, zdali nejnovější ovladač bude podporovat ovládání i druhého větráku. Nicméně i tato nejnovější verze ovládání dalšího větráku nepodporuje.

## ZÁVĚR V ANGLIČTINĚ

### CONCLUSION IN ENGLISH

The installation of *Linux* server was the fundamental part of the Bachelor Thesis. It was essential to decide what distribution of the operating system *Linux* would be used, what distribution would be the most fitting for *NAS* configuration. Before the installation it was decided to choose *Debian* distribution. In comparison to *Fedora* distribution, *Debian* has got much more developed installation concerning the creation of *RAID* and configuration of *Logical Volume Management*.

When installing *Linux*, the first problem was caused by the absence of *CD* drive in the server chassis. This chassis was not adjusted for connection of internal drive, for that reason it was necessary to choose another way of installation. One of possible options was to use an external drive, but finally another one was chosen. This alternative was to create a boot USB flash disk that was done by program *UNetbootin*. Even though there are numbers of similar programs, *UNetbootin* was picked mainly for the reason that it is not necessary to install it and it is possible to use it immediately after its download.

During the subsequent installation it was required to decide which direction to go at the disk distribution and choice of the proper *RAID* type. Considering the fact that the server was set up by the double *SATA* disks, there were possible only two variants for the selection of *RAID* type: *RAID 0* or *RAID 1*. *RAID 1* was selected, when the whole capacity of both disks was not used, but the priority was to prevent the servers from their failure. In the other words, the data safety and stability security was the priority, which is really very important in the case of servers.

After that in the part of the disk distribution there was chosen the option of creation of *volume group* through the whole *RAID 1*. It is the best way for the subsequent enlargement of disk capacity and from the experience it can be said that it is the most used method of all in the real IT world. If two other disks were added into currently used server, (it is possible to execute this, because there are four slots for hard disks in the server chassis), it would be very easy to enlarge disk capacity of already created *volume group*. It is one of the most suitable option for capacity management and capacity enlargement.

The follow-up part was focused on the configuration of *NFS* and *SMB* protocols. The configuration of *SMB* is partially difficult, nevertheless an administrator benefits from security of this protocol. *NFS* has a lower level of security, even though the configuration is running without problems. For that reason it is recommended to share data by *SMB* protocols for the server determined to data sharing in larger companies. Setting of accesses for various groups of users and individual users is very detailed in this protocol.

The next step was the implementation of *iSCSI* support. Network protocol *iSCSI* is not suitable for data sharing among more clients and the main reason is that there is more complicated configuration on the client's side and minimal possibility of security on the server side. Protocol *SMB* is more suitable for sharing data of more clients. However protocol *iSCSI* is appropriate for use within server virtualization, when the whole hierarchy of *VMware* is mostly based on *iSCSI*. Possible use of *iSCSI* is in the situation, when there are created two types of virtual network, external and internal network. It is convenient to use *iSCSI* in the internal network for interconnection of ESX server and shared file storages. Furthermore it is possible to use it for back up of virtual servers by the means of *Veeam Backup and Replication*. *iSCSI* is used as well in the cases when it is impossible to use the communication by way of *Fibre Channel* because of financial or technical insufficiency.

When evaluating the proposal to security, it is required to consider the environment in which the server will be used. It is important to know, if the server is accessible only within the internal network or if the server would share data by way of internet. The most dangerous way of data sharing is via internet, when the server is likely to be attempted to put out somehow, e.g. by way of *DoS* attack. Mainly for that reason the restriction was selected, the restriction to *ROOT* users' login by way of *SSH* and the restriction to the configuration of *Fail2ban* for blocking *IP* address after mistaken server login.

Considering the question of setting for power management, it is essential to mention that whatever attempt to configuration of power management must be supported by used hardware. It proved true in the case of this thesis during the trial of the scaling configuration to the frequency of the processor. The processor, used here, Intel Atom D525 did not support this function so in order to lower the electricity load we approached to fixed lowering of the processor frequency. Regarding the fact that the thesis is about *NAS* configuration, where the data sharing via computer network is supposed, the fixed setting of processor frequency is not assumed as a big problem. This kind of choice would not be

suitable for the case when the server applications need full numeral capacity, even only a few minutes per day. In such a case it would be appropriate to use scaling of processor frequency, when the processor frequency is changed due to the demands of individual applications. Of course another type of the processor is needed, the type that supports this function.

In addition to scaling of processor, the option of fan adjusting was chosen. When setting a fan, it is appropriate to set a thermistor of the processor, where the changes of the temperature are the most common. It is recommended to adjust only rotations of some fans, because if a thermistor gets damaged and does not react to the temperature change, it may end in damage of some inner hardware components of the server. During the configuration a problem occurred. The operating system driver enabled to control only one fan and it was *CPU FAN*.

At the period of finishing this Bachelor Thesis, a new version of operating system *Linux Debian 7.0* was issued. There was a special testing of this new version concerning power management. The test concerned finding if the version supported controlling another fan or not. Nevertheless this version does not support this function too.

**SEZNAM POUŽITÉ LITERATURY**

- [1] NAS. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-02]. Dostupné z: <http://cs.wikipedia.org/wiki/NAS>
- [2] ČERNÝ, Jiří. NAS vs. SAN - jak na správu dat?. In: [online]. [cit. 2013-05-05]. Dostupné z: <http://www.svethardware.cz/nas-vs-san-jak-na-spravu-dat/27556>
- [3] MRNKA, Ladislav. Network attached storage (NAS).  
In: *Network Storage* [online]. [cit. 2013-05-02]. Dostupné z: <http://www.kiv.zcu.cz/~simekm/vyuka/pd/zapocty-2004/san-mrnka/nas.html>
- [4] Non-RAID drive architectures. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-05]. Dostupné z: [http://en.wikipedia.org/wiki/Non-RAID\\_drive\\_architectures](http://en.wikipedia.org/wiki/Non-RAID_drive_architectures)
- [5] RAID. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-02]. Dostupné z: <http://cs.wikipedia.org/wiki/RAID>
- [6] STANEK, William R. *Mistrovství v Microsoft Windows Server 2008: [kompletní informační zdroj pro profesionály]*. Vyd. 1. Brno: Computer Press, 2009, 1364 s. ISBN 978-80-251-2158-0.
- [7] Co je to RAID?. In: [online]. [cit. 2013-05-05]. Dostupné z: [http://www.netcorp.cz/storage\\_systems/raid\\_description.htm](http://www.netcorp.cz/storage_systems/raid_description.htm)
- [8] RUSSEL, Charlie. *Microsoft windows server 2003: velký průvodce administrátora*. Vyd. 1. Brno: Computer Press, 2005, 1374 s. ISBN 80-251-0579-2.

- [9] LOWE, Scott. *Mistrovství ve VMware vSphere 4: [kompletní průvodce profesionální virtualizací]*. Vyd. 1. Brno: Computer Press, 2010, 662 s. Mistrovství. ISBN 978-80-251-2915-9.
- [10] KERSLAGER, Milan. LVM v *Linuxu*. In: [online]. [cit. 2013-05-02]. Dostupné z: [http://www.pslib.cz/ke/LVM\\_v\\_Linuxu](http://www.pslib.cz/ke/LVM_v_Linuxu)
- [11] SHAH, Steve. *Administrace systému Linux: překlad čtvrtého vydání*. 1. vyd. Praha: Grada, 2007, 426 s. ISBN 978-80-247-1694-7.
- [12] A simple introduction to working with LVM. In: [online]. [cit. 2013-05-02]. Dostupné z: <http://www.debian-administration.org/articles/410>
- [13] MATÝSEK, Miroslav. *Počítačové sítě: UČEBNÍ PREZENTACE UTB FAI UAI*.
- [14] DOSTÁLEK, Libor. *Velký průvodce protokoly TCP/IP: bezpečnost*. Vyd. 1. Praha: Computer Press, 2001, xvi, 565 s. ISBN 80-722-6513-X.
- [15] BIGELOW, Stephen J. *Mistrovství v počítačových sítích: správa, konfigurace, diagnostika a řešení problémů*. Vyd. 1. Překlad Petr Matějů. Brno: Computer Press, 2004, 990 s. ISBN 80-251-0178-9.
- [16] ECKSTEIN, Robert, Peter KELLY a David COLLIER-BROWN. *Samba: Linux - souborový a tiskový server pro heterogenní síť*. Vyd. 1. Praha: Computer Press, 2001, xvi, 378 s. ISBN 80-722-6463-X.
- [17] Samba (software). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-02]. Dostupné z: [http://cs.wikipedia.org/wiki/Samba\\_\(software\)](http://cs.wikipedia.org/wiki/Samba_(software))

- [18] Network File System. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-02]. Dostupné z: [http://cs.wikipedia.org/wiki/Network\\_File\\_System](http://cs.wikipedia.org/wiki/Network_File_System)
- [19] PETRŽELA, Radim. Fenomén iSCSI: aneb Lepší iSCSI v hrsti nežli FC na střeše!. In: [online]. [cit. 2013-05-05]. Dostupné z: <http://www.datavpeci.cz/webdvp.nsf/articles/07ABA713F282B84CC125750C00471D9D>
- [20] NEMETH, Evi, Garth SNYDER a Trent R HEIN. *Linux: kompletní příručka administrátora : 2. aktualizované vydání*. Vyd. 1. Brno: Computer Press, 2008, 984 s. ISBN 978-80-251-2410-9.
- [21] ČEVELA, Lubomír. Samba - Tanec s Okny: smb.conf, nastavení klientů. In: [online]. [cit. 2013-05-05]. Dostupné z: <http://www.root.cz/clanky/samba-smb-conf-nastaveni-klientu/>
- [22] Pripojenie iSCSI disku. In: *AbcLinuxu* [online]. [cit. 2013-05-07]. Dostupné z: <http://www.abclinuxu.cz/faq/disky/pripojenie-iscsi-disku>
- [23] VESELSKÝ, J. *LINUX: dokumentační projekt*. Vyd. 2. Brno: Computer Press, 2003, 1001 s. ISBN 80-722-6761-2.
- [24] HOWTO fail2ban 0.7.x. In: [online]. [cit. 2013-05-06]. Dostupné z: [http://www.fail2ban.org/wiki/index.php/HOWTO\\_fail2ban\\_0.7.x](http://www.fail2ban.org/wiki/index.php/HOWTO_fail2ban_0.7.x)
- [25] KRČMÁŘ, Petr. Jak na uspávání disku v Linuxu?. In: [online]. [cit. 2013-05-06]. Dostupné z: <http://www.root.cz/clanky/jak-na-uspavani-disku-v-linuxu/>

- [26] NONDEK, Luboš. Cpufreq - dynamické škálování frekvence procesoru. In: [online]. [cit. 2013-05-06]. Dostupné z: <http://www.abclinuxu.cz/clanky/system/cpufreq-dynamicke-skalovani-frekvence-procesoru>
- [27] Enabling power management on Debian. In: [online]. [cit. 2013-05-06]. Dostupné z: [http://blog.peacon.co.uk/wiki/Enabling\\_power\\_management\\_on\\_Debian](http://blog.peacon.co.uk/wiki/Enabling_power_management_on_Debian)
- [28] REINER, Marius. Fancontrol(8). In: *Die.net* [online]. [cit. 2013-05-08]. Dostupné z: <http://linux.die.net/man/8/fancontrol>
- [29] Regulace otáček větráčků – fancontrol. In: *AbcLinuxu* [online]. [cit. 2013-05-08]. Dostupné z: <http://www.abclinuxu.cz/blog/ritchie/2006/3/regulace-otacek-vetracku-ndash-fancontrol>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

ATA	Advanced Technology Attachment
BIOS	Basic Input-Output System
CD	Compact Disc
CIFS	Common Internet File System
CPU	Central Processing Unit
DARPA	Defense Advanced Research Projects Agency
DNS	Domain Name System
DOD	Department of Defense
EXT	Second Extended Filesystem
FCP	Fibre Channel Protocol
FDDI	Fiber Distributed Data Interface
FTP	File Transfer Protocol
GB	Giga Byte
HDD	Hard Disk Drive
CHAP	Challenge Handshake Authentication Protocol
IBM	International Business Machines Corporation
ICMP	Internet Control Message Protocol
IDE	Integrated Drive Electronics
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSec	Internet Protocol Security
IPV4	Internet Protocol version 4
iSCSI	Internet Small Computer System Interface
IT	Informační Technologie

---

LAN	Local Area Network
LLC	Logical Link Control
LUN	Logical Unit Number
LV	Logical Volume
MTA	Mail Transfer Agent
NAS	Network Attached Storage
NFS	Network File System
NT	New Technology
NTP	Network Time Protocol
OSI	Open Systems Interconnection
PC	Personal Computer
PDU	Protocol Data Unit
PV	Physical Volume
RAID	Redundant Array of Inexpensive/Independent Disks
SAN	Storage Area Network
SCSI	Small Computer System Interface
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
TCP/IP	Transmission Control Protocol/Internet
UDP	User Datagram Protocol
UNC	Uniform Naming Convention
USB	Universal Serial Bus
UTB	Univerzita Tomáše Bati
VG	Volume Group

## SEZNAM OBRÁZKŮ

<i>Obr. 1. NAS úložiště [1]</i> .....	11
<i>Obr. 2. Topologie NAS [2]</i> .....	12
<i>Obr. 3. Topologie NAS [2]</i> .....	13
<i>Obr. 4. RAID 0 [7]</i> .....	15
<i>Obr. 5. RAID 1 [7]</i> .....	16
<i>Obr. 6. RAID 0+1 [7]</i> .....	17
<i>Obr. 7. RAID 5 [7]</i> .....	18
<i>Obr. 8. RAID 6 [7]</i> .....	19
<i>Obr. 9. Skladba LVM [12]</i> .....	20
<i>Obr. 10. Vrstvy modelu OSI [13]</i> .....	21
<i>Obr. 11. Rozdíl mezi vrstvami OSI a TCP/IP modelu [13]</i> .....	22
<i>Obr. 12. Příklad sběrnice topologie [13]</i> .....	24
<i>Obr. 13. Příklad sítě 10Base2T [13]</i> .....	25
<i>Obr. 14. Zapouzdření iSCSI v TCP/IP a ethernetových rámcích [9]</i> .....	28
<i>Obr. 15. Výběr jazyka instalace</i> .....	33
<i>Obr. 16. Volba časového pásma</i> .....	33
<i>Obr. 17. Konfigurace RAID 1</i> .....	34
<i>Obr. 18. Vytvoření volume group</i> .....	34
<i>Obr. 19. Logické volume</i> .....	35
<i>Obr. 20. Volume group „source“</i> .....	36
<i>Obr. 21. LV ROOT</i> .....	36
<i>Obr. 22. SSH služba status</i> .....	37
<i>Obr. 23. Server NFS</i> .....	38
<i>Obr. 24. Konfigurace /etc/exports</i> .....	40
<i>Obr. 25. Příkaz df -h</i> .....	41
<i>Obr. 26. Výpis /etc/passwd</i> .....	48
<i>Obr. 27. Instalace iSCSI target-dkms</i> .....	50
<i>Obr. 28. iSCSI target enable</i> .....	51
<i>Obr. 29. iSCSI status</i> .....	51
<i>Obr. 30. Cílový portál iSCSI</i> .....	53
<i>Obr. 31. Připojený target</i> .....	53
<i>Obr. 32. Diskový prostor</i> .....	53

---

<i>Obr. 33. Kontrolní výstup</i> .....	54
<i>Obr. 34. Virtuální disk</i> .....	55
<i>Obr. 35. Nepovolený přístup</i> .....	57
<i>Obr. 36. Přihlášení k serveru pře SSH jako uživatel UTB</i> .....	57
<i>Obr. 37. Příkaz su</i> .....	58
<i>Obr. 38. Fai2Ban.log</i> .....	59
<i>Obr. 39. Informace o HDD</i> .....	62
<i>Obr. 40. Kontrola hdparm.sh</i> .....	63
<i>Obr. 41. Kontrolní výpis</i> .....	65
<i>Obr. 42. Příkaz sensors</i> .....	66
<i>Obr. 43. Konfigurace /etc/fancontrol</i> .....	67
<i>Obr. 44. Výpis příkazu fanconrol</i> .....	68