

Virtuální laboratoř mikropočítačů se vzdáleným přístupem

Virtual laboratory of microcontrollers with remote access

Bc. Věra Budíková

Diplomová práce
2011



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Věra BUDÍKOVÁ**
Osobní číslo: **A09467**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Virtuální laboratoř mikropočítačů se vzdáleným přístupem**

Zásady pro vypracování:

S pomocí virtualizace implementujte vzdálený přístup k pracovištím laboratoře mikropočítačů.

Zaměřte se na následující problémy:

- 1. Prostudujte dostupné virtualizační technologie a porovnejte jejich výkon.**
- 2. Analyzujte možnosti realizace sdílených diskových polí.**
- 3. Navrhněte způsob připojení stávajících USB měřících přístrojů a vývojových desek přes Ethernet.**
- 4. Implementujte zabezpečení vzdáleného přístupu.**
- 5. Realizujete několik příkladů laboratorních úloh vhodných pro virtuální laboratoř.**

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. RUEST, Danielle; RUEST, Nelson. **Virtualizace: Podrobný průvodce**. 1. vyd. Brno: Computer Press, 2010. 408 s. ISBN 978-80-251-2676-9.
2. THOMAS, Thomas M. **Zabezpečení počítačových sítí: bez předchozích znalostí**. 1. vyd. Brno: Computer Press, 2005. 344 s. ISBN 80-251-0417-6.
3. Kolektiv autorů. **Linux: Dokumentační projekt**. 4. vyd. Brno: Computer Press, 2008. 1336 s. ISBN 978-80-251-1525-1.
4. KRČMÁŘ, Petr. **Linux: postavte si počítačovou síť**. 1. vyd. Praha: Grada, 2008. 184 s. ISBN 978-80-247-1290-1.
5. KRÁL, Jiří. **Řešené příklady ve VHDL: Hradlová pole FPGA pro začátečníky**. 1.vyd. Praha: BEN – technická literatura, 2010. 128 s. ISBN 978-80-730-0257-2.

Vedoucí diplomové práce:

Ing. Tomáš Dulík

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

24. února 2011

Termín odevzdání diplomové práce:

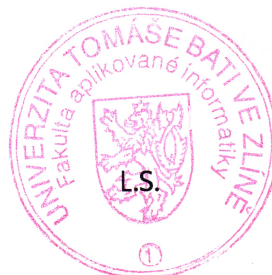
18. května 2011

Ve Zlíně dne 24. února 2011



prof. Ing. Vladimír Vašek, CSc.

děkan



doc. Mgr. Roman Jašek, Ph.D.

ředitel ústavu

ABSTRAKT

Diplomová práce se zabývá projektem Virtuální laboratoře aplikace mikroprocesorové techniky (VLAM). Cílem celého projektu bylo implementovat virtuální WWW laboratoř pro vývoj vestavěných systémů, navrhnout metodologii a SW nástroje pro návrh těchto systémů nové generace. V teoretické části je stručně popsána daná problematika, podrobněji jsou rozvedena témata, na kterých jsem se přímo podílela. V praktické části jsou popsána jednotlivá řešení a jejich testování, tedy výběr virtualizační technologie, realizace sdílených diskových polí, návrh způsobu připojení USB přes IP a implementace bezpečnosti vzdáleného přístupu.

Klíčová slova: VLAM, virtualizace, USB přes IP, VPN, RAID.

ABSTRACT

The thesis deals with a project focused on a Virtual laboratory of microprocessor technology application (VLAM). The aim of the whole project was to implement a virtual WWW laboratory for inbuilt systems development, to design a methodology and SW tools for systems of new generation designing. In the theoretical part there is briefly described the problematics, the topics I was involved in are described in more details. In the practical part there are described the solutions and their testing, such as a selection of virtualization technology, a realization of shared disc arrays, a design of methods for USB over IP connections and an implementation of security of remote access.

Keywords: VLAM, virtualization, USB over IP, VPN, RAID.

Úvodem bych ráda poděkovala svému vedoucímu práce, panu Ing. Tomáši Dulíkovi, za cenné rady, pomoc a trpělivost při tvorbě této diplomové práce. Dále bych chtěla poděkovat svému příteli za podporu a také mnoha dalším lidem, kteří svou radou nebo jakýmkoliv jiným způsobem přispěli k vytvoření této práce.

Tato práce byla podpořena grantem MŠMT č. 2C06008 „VLAM: Virtuální laboratoř aplikace mikroprocesorové techniky“, který byl schválen v rámci programů na podporu výzkumu a vývoje - Národní program výzkumu II.

Motto:

Otevřete dveře, odcházím žít.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/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 diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/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 diplomovou/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 – diplomovou/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 diplomové/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 diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/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 diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ABSTRAKT	4
ABSTRACT	4
ÚVOD	10
I TEORETICKÁ ČÁST	11
1 ANALÝZA PROBLÉMU	12
1.1 80C537 MICROCONTROLLER REMOTE LAB FOR E-LEARNING TEACHING	13
1.2 INTERNATIONAL COLLABORATIVE LEARNING USING REMOTE WORKBENCHES FOR 8-BIT MICROCONTROLLER COURSES	14
1.3 INTERNET SCHOOL EXPERIMENTAL SYSTEM - ISES.....	14
1.4 SHRNUÍ.....	15
2 ARCHITEKTURA LABORATOŘE VLAM	16
2.1 SERVER PRO VIRTUALIZACI	16
2.1.1 Virtualizace.....	16
2.2 HW/SW KONFIGURACE SERVERU.....	17
2.2.1 Konfigurace serveru	18
2.2.2 Softwarová konfigurace	18
2.2.3 Zabezpečení vzdáleného přístupu.....	18
2.3 USB PŘES IP.....	19
2.4 DISKOVÉ POLE.....	19
2.4.1 Výhody a nevýhody FakeRAIDu a Linux SW RAIDu.....	20
2.5 SW/HW SERVERU VLAM.....	22
2.5.1 Vstupní brána projektu (VLAM Gateway).....	22
2.6 VLAXICON.....	24
2.7 TECHNOLOGIE HRADLOVÝCH POLÍ.....	25
2.7.1 Spartan-3E Starter Kit	26
2.7.2 Výukové lekce pro PicoBlaze.....	27
2.8 VLAM IDE.....	27
2.8.1 Integrace nástrojů.....	27
2.9 PICOBLAZE C COMPILER (PBCC).....	29
2.9.1 Historie.....	29
2.9.2 Architektura.....	29
II PRAKTICKÁ ČÁST	30
3 VIRTUALIZACE 8 WINDOWS XP HOSTŮ NA LINUXOVÉM SERVERU	30
3.1 POUŽITÝ HARDWARE.....	30
3.2 VÝBĚR VIRTUALIZAČNÍCH NÁSTROJŮ.....	30
3.3 VÝBĚR DISTRIBUCE LINUXU.....	31

3.4	PŘÍSTUP K VIRTUÁLNÍM STROJŮM.....	31
3.5	VÝBĚR ZÁTĚŽOVÝCH TESTŮ.....	31
3.5.1	Testovací skripty.....	32
3.6	VIRTUALIZAČNÍ NÁSTROJE – ZÁKLADNÍ KONFIGURACE.....	33
3.6.1	Xen.....	33
3.6.2	KVM.....	35
3.6.3	VMware Server.....	36
3.6.4	VMware ESX Server.....	36
3.7	TVORBA VIRTUÁLNÍCH STROJŮ.....	36
3.7.1	Správa virtuálních strojů.....	37
3.8	VÝSLEDKY TESTŮ: KVM VERSUS XEN.....	40
3.8.1	Pro jeden virtuální stroj.....	40
3.8.2	Pro osm virtuálních strojů.....	40
3.8.3	Subjektivní porovnání z pohledu uživatele.....	41
3.9	SHRNUTÍ VÝSLEDKŮ.....	41
4	REALIZACE SDÍLENÝCH DISKOVÝCH POLÍ.....	43
4.1	POUŽITÝ HARDWARE.....	43
4.2	POSTUP TESTOVÁNÍ.....	43
4.3	BONNIE++.....	43
4.4	FAKE RAID.....	44
4.5	SOFTWAREVÝ RAID.....	44
4.6	HOT-PLUG.....	44
4.7	VÝSLEDKY TESTOVÁNÍ.....	47
5	USB OVER IP: SOFTWAREVÉ ŘEŠENÍ.....	48
5.1	POUŽITÝ HARDWARE.....	49
5.2	POUŽITÉ OPERAČNÍ SYSTÉMY.....	51
5.3	SOFTWARE VHODNÝ PRO ŘEŠENÍ USB OVER IP PROBLÉMU.....	51
5.4	SOFTWARE POUŽITÝ K OVLÁDÁNÍ PŘIPOJENÝCH ZAŘÍZENÍ.....	52
5.5	TESTOVÁNÍ JEDNOTLIVÝCH ŘEŠENÍ.....	53
5.5.1	USB/IP project.....	53
5.5.2	USB Server & USB Client for Linux.....	56
5.5.3	USB over Network.....	56
5.5.4	USB to Ethernet Connector.....	57
5.5.5	Celkové výsledky.....	57
5.6	SHRNUTÍ.....	57
6	USB OVER IP: HARDWAROVÉ ŘEŠENÍ.....	59
6.1	POUŽITÝ HARDWARE.....	59
6.2	POSTUP TESTOVÁNÍ.....	59
6.3	VÝSLEDKY TESTOVÁNÍ.....	60

7	ZABEZPEČENÍ VZDÁLENÉHO PŘÍSTUPU.....	61
7.1	FREERADIUS.....	61
7.1.1	Instalace a konfigurace.....	61
7.2	PPTP VPN.....	64
7.2.1	PPTP – server.....	64
7.2.2	PPTP - klient.....	65
7.2.3	Výsledky.....	67
7.3	OPENVPN.....	67
7.3.1	Instalace:.....	67
7.3.2	Vytvoření SSL/TLS certifikátů:.....	67
7.3.3	Konfigurace PAM modulu pro přihlašování.....	68
7.3.4	Konfigurace OpenVPN serveru.....	69
7.3.5	Klient – MS Windows XP.....	69
7.3.6	Klient - Linux.....	70
	ZÁVĚR.....	71
	CONCLUSION.....	72
	SEZNAM POUŽITÉ LITERATURY.....	73
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	76
	SEZNAM OBRÁZKŮ.....	78
	SEZNAM TABULEK.....	80

ÚVOD

V rámci svojí diplomové práce jsem se podílela na řešení projektu Virtuální laboratoře aplikace mikroprocesorové techniky (VLAM), který je zaměřen na aplikovaný výzkum metodologie SW nástrojů a technologie virtuálního WWW serveru dovolujícího automatizovat generování, konfiguraci a kompilaci SW i HW komponent pro systémy s procesorovým jádrem a specializovanými HW periferiemi, které mohou být na FPGA.

Cílem celého projektu bylo implementovat virtuální WWW laboratoř pro vývoj vestavěných systémů, navrhnout metodologii a SW nástroje pro návrh těchto systémů nové generace. Dále navržení několika demonstračních specializovaných HW periferií usnadňující implementaci vestavných systémů a tyto nástroje a periferie zpřístupnit široké mezinárodní odborné veřejnosti. Nástroje podporují řadu mikroprocesorů, periferií a jejich abstrakcí tak, aby dovolovaly návrhářům společný vývoj vestavných systémů v rámci virtuální WWW laboratoře.

Architektura VLAM poskytuje přístup k jednotlivým pracovištím laboratoře pomocí virtuálních strojů, které jsou spojeny s laboratorním zařízením přes ethernet nebo pomocí převodníků „ethernet-to-USB“. Struktura zapojení a jednotlivé komponenty jsou popsány v podkapitolách 2.1-2.4. V rámci návrhu architektury jsem provedla výběr, testování, zprovoznění virtualizačního softwaru, řešení přenosu USB přes Ethernet a testování chování diskových polí v Linuxu.

Kapitola 2.5 se věnuje systému řízení laboratoře, který je pojmenován VLAM Gateway, kde jsem přispěla realizací zabezpečení pomocí VPN.

V poslední kapitole pro úplnost zmiňuji část řešení, na kterém jsem se nepodílela, což jsou softwarové aplikace, které běží na studentských virtuálních strojích.

I. TEORETICKÁ ČÁST

1 ANALÝZA PROBLÉMU

Vedení výuky mikroprocesorové techniky doposud probíhalo v laboratořích, kde studenti fyzicky pracovali s vývojovými kity a dalšími zařízeními (osciloskop, generátor, aj.). Pokud měl student zájem pracovat i mimo výuku, musel se domluvit s vyučujícím, který ho do laboratoře pustil a po odchodu studenta musel zkontrolovat stav pracoviště. Některé školy řeší výuku pomocí půjčování vývojových kitů studentům domů, což je však finančně náročné. Dochází také k větší poruchovosti zařízení a tím ke zvýšení dalších nákladů z důvodů oprav.

Řešení projektu virtuální laboratoře umožňuje studentům pracovat z domu stejně, jakoby byli přítomni přímo v laboratoři, z čehož vyplývá několik výhod nejen pro samotné studenty, ale také pro školu. Studenti nejsou vázáni na dobu otevření školy ani na hledání pověřené osoby, která by kontrolovala jejich činnost v laboratoři mimo hodiny výuky. Virtuální laboratoř funguje v režimu 24/7/365. Student nemusí mít nainstalován speciální software pro práci s jednotlivými zařízeními, vše se nachází na virtuálním stroji, na který se studenti připojují. Z pohledu školy je jistě výhodná nabídka nového způsobu vedení výuky atraktivní pro studenty. Dále jsou to však také finanční aspekty:

- Není potřeba pověřená osoba, která by v laboratoři dohlížela na studenty.
- Poruchovost zařízení je menší, než při přímém kontaktu se studentem.
- Správa pracovních stanic je méně náročná.
- Šetření místa a energie díky menšímu množství použitého hardwaru.

Virtuální laboratoř má také své nevýhody:

- Při výpadku sítě či hardwarové poruše na serveru je práce znemožněna.
- Vyšší pořizovací náklady.
- Student ztratí možnost fyzického kontaktu se zařízením.

Tab. 1: Ekonomický rozbor, 1 lab. pracoviště = 1 plnohodnotné PC. [1]

	Nonstop provoz	Běžný provoz
Roční náklady na elektřinu (12 PC)	67 277 Kč	16 589 Kč
Roční náklady na 1 laboratoř (12 PC)	119 237 Kč	68 549 Kč
Roční náklady na pořízení a správu všech PC fakulty (300ks)	1 299 000 Kč	
Roční náklady na elektřinu všech PC (300ks)	1 681 920 Kč	414 720 Kč
Celkové roční náklady na všechna PC	2 980 920 Kč	1 713 720 Kč

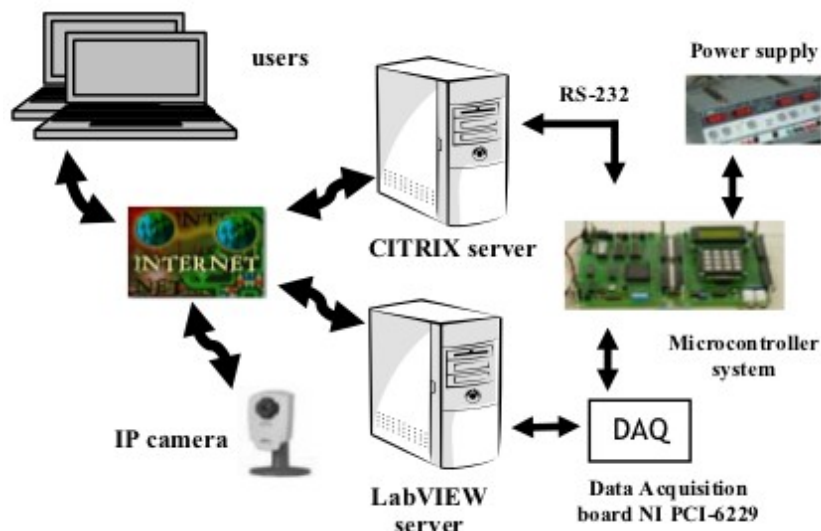
Tab. 2: Ekonomický rozbor, plně virtualizovaná laboratoř. [1]

	Nonstop provoz	Běžný provoz
Roční náklady na elektřinu (12 PC)	25 229 Kč	6 221 Kč
Roční náklady na 1 laboratoř (12 PC)	36 840 Kč	43 061 Kč
Roční náklady na pořízení a správu všech PC fakulty (300ks)	921 000 Kč	
Roční náklady na elektřinu všech PC (300ks)	630 720 Kč	155 520 Kč
Celkové roční náklady na všechna PC	1 551 720 Kč	1 076 520 Kč

Samozřejmě řešení virtuální laboratoře podobné VLAMu není jedinečné. V různých obměnách se lze setkat s několika projekty. Zde je uvedeno několik příkladů.

1.1 80C537 Microcontroller Remote Lab for E-Learning Teaching

Projekt Katalánské technické univerzity a Koruntanské univerzity aplikovaných věd, umožňuje studentům práci z domova. Základem je mikrokontrolér μ de537 a kompletní profesionální vývojové prostředí pro jednočipové mikroprocesory, Keil μ Vision2. Mikrokontrolér je spojen se serverem sériovou linkou RS-232. K nahrávání programů do mikrokontroléru a debugování je potřeba Keil μ Vision2. Ke vzdálenému přístupu k tomuto softwaru je využit Citrix Application Server software, který poskytuje rozsáhlému počtu klientů bezpečný přístup k aplikacím a programům nainstalovaných na serveru, proto není potřeba mít Keil μ Vision2 nainstalovaný na klientském počítači. K monitorování LCD a LED se využívá IP kamera. HUMAN BOARD INTERFACE slouží ke kontrole, monitorování a správě dat a umožňuje uživateli chovat se tak, jako by byl přítomný přímo v laboratoři. Vyvinut byl pomocí LabVIEW. [2]



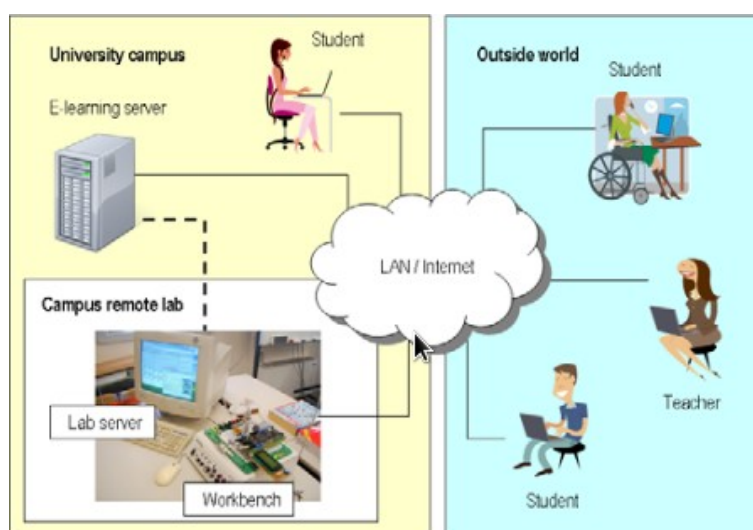
Obr. 1: Schéma projektu - 80C537 Microcontroller Remote Lab for E-Learning Teaching. [2]

1.2 International collaborative learning using remote workbenches for 8-bit microcontroller courses

Projekt Portské univerzity v Portugalsku a Univerzity jižní Austrálie.

Vzdálené pracoviště:

- E-learning server Moodle
- Laboratorní server, který spravuje pracoviště.
- Samotné fyzické pracoviště, obsahující NI ELVIS pracovní stanici a 80C51 mikrokontrolér.
- Software:
 - Keil μ Vision
 - LabVIEW [3]



Obr. 2: Schéma projektu - 3.2 International collaborative learning using remote workbenches for 8-bit microcontroller courses .[3]

1.3 Internet School Experimental System - iSES

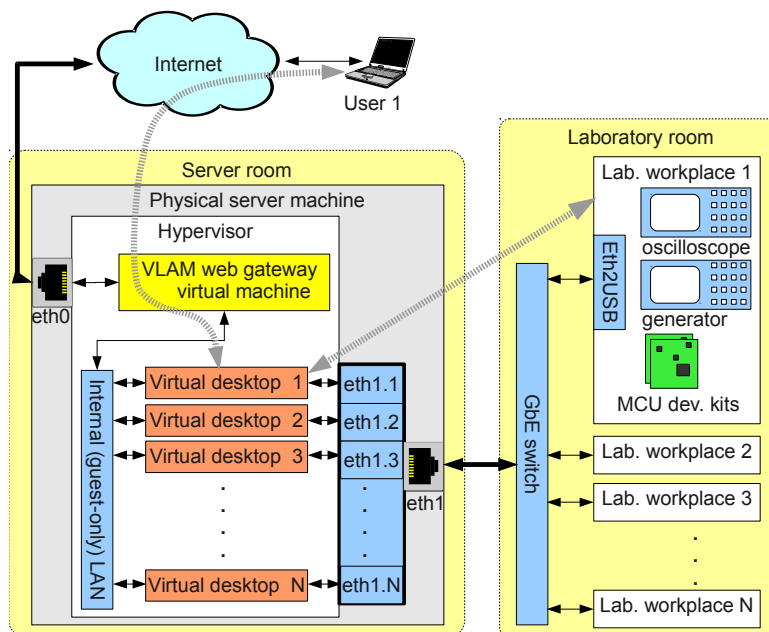
Měřicí a laboratorní studio iSES je široká otevřená platforma, která umožňuje měření a řízení experimentů. Systém iSES je otevřený modulární univerzální měřicí systém pro fyziku, chemii, biologii, elektrotechniku, elektroniku, automatizaci, měření, aj. Celkem je k dispozici 20 čidel (teploměr, ampérmetr, voltmetr, siloměr, snímač polohy, optická závora, mikrofon, manometr, pH metr, konduktometr, relé, reproduktor, sonar, ohmmetr, měřič kapacit, snímač srdečního tepu aj. Čidla se sama autodetekují, mají lineární charakteristiku

a pokud je to možné, tak diferenciální vstupy. Souprava iSES je český výrobek, který se vyrábí již 25 let a postupně se vyvíjí, tak jak se vyvíjejí počítače a operační systém. [4]

1.4 Shrnutí

Zmíněná první dvě řešení jsou si poměrně podobná, jelikož obě používají stejnou softwarovou základnu: Keil μ Vision a LabVIEW. Oba produkty jsou finančně poměrně náročné. Na rozdíl od projektu VLAM, se jedná o vzdálený přístup k jednomu mikrokontroléru. Ve VLAM se jedná o několik zařízení včetně osciloskopů a generátorů, tedy o mnohem složitější strukturu než u zmíněných prvních dvou řešení. Projekt VLAM je řešen open-source a je zopakovatelný v kterékoliv učebně s minimálními náklady. Třetí zmíněné řešení je velice zajímavé a rozsáhlé, výhodou je jeho modulárnost.

2 ARCHITEKTURA LABORATOŘE VLAM



Obr. 3: Architektura projektu VLAM. [5]

2.1 Server pro virtualizaci

Server pro virtualizaci je základní komponentou infrastruktury VLAM. V následujících odstavcích je obsažen jeho stručný popis.

2.1.1 Virtualizace

Virtualizace je proces, kdy se v jednom operačním systému spustí pomocí virtualizačního systému jiný operační systém. [6]

Výhody virtualizace:

- Efektivní využití zdrojů.
- Snížení nákladů na provoz.
- Zmenšení prostorových nároků na hardware.
- Jednodušší správa a údržba.
- Centralizované diskové úložiště, zvýšená spolehlivost – jednodušší možnost zálohování.
- Oddělenost hostovaných systémů.

Nevýhody virtualizace

- V případě výpadku sítě či hardwaru nepřístupnost pracovišť.

Typy virtualizace

- Emulace - virtualizace hardwarových komponent za účelem simulace jiné hardwarové platformy, nevyužívá se hardwarové podpory virtualizace. Virtualizace je pouze softwarová. Emulace umožňuje programům běžet na jiné platformě, než pro jakou jsou naprogramovány (např. QEMU, DOSBox).
- Hardwarově asistovaná virtualizace – využití speciálně navržených CPU či jiných hardwarových komponent ke zvýšení výkonu hostovaných systémů (např. KVM, VirtualBox, VMware Server).
- Paravirtualizace - virtuální stroj nesimuluje hardware, ale místo toho nabízí speciální aplikační rozhraní (API), které vyžaduje určité modifikace hostovaného operačního systému, aby mohl být tento OS nad virtuálním strojem spouštěn (Xen).
- Virtualizace na úrovni OS - virtualizační metoda, ve které jádro OS umožňuje běh více izolovaných prostředí (kontejnerů, VPS) namísto jediného, jako je tomu u běžných OS (např. OpenVZ, FreeBSD jails).
- Aplikační virtualizace – virtualizace poskytující pouze základní komponenty potřebné pro běh aplikace (soubory, registry aj.). Aplikační virtualizace dosahuje rychlejšího běhu aplikace než virtualizace aplikace společně s celým OS, zpravidla je však méně dokonalá. V některých případech je možné dosáhnout i rychlejšího chodu aplikace než v nativním OS (Wine). [7]

2.2 HW/SW konfigurace serveru

Pro projekt VLAM bylo potřeba stanovit požadavky na HW a software konfiguraci serveru a specifikovat jednotlivé HW/SW komponenty vhodné k naplnění daných požadavků.

Sever pro projekt VLAM je použit jako aplikační server poskytující vzdálený přístup k sadě publikovaných aplikací sloužících pro nácvik vývoje embedded software na konkrétních vývojových kitech. Server umožňuje vzdálený přístup k virtuálním, vzájemně odděleným pracovním stanicím. Přihlášení k jednotlivým stanicím je řešeno prostřednictvím zabezpečeného webového rozhraní. Každá virtuální pracovní stanice obsahuje softwarové vybavení pro testování a vývoj připojeného HW embedded zařízení (vývojového kitu). Aktuální stav vývojového kitu je monitorován jednak prostřednictvím SW komponent vývojového prostředí a jednak pomocí streamovaného videa umožňujícího sledovat chování systému v reálném čase. Pro provoz publikovaných pracovních stanic se využívá virtualizační technologie KVM. Výběr vhodného virtualizačního řešení předcházela řada testů, které jsou popsány v praktické části diplomové práce

2.2.1 Konfigurace serveru

Server SuperMicro S1420Q-M4V SYS .

2.2.2 Softwarová konfigurace

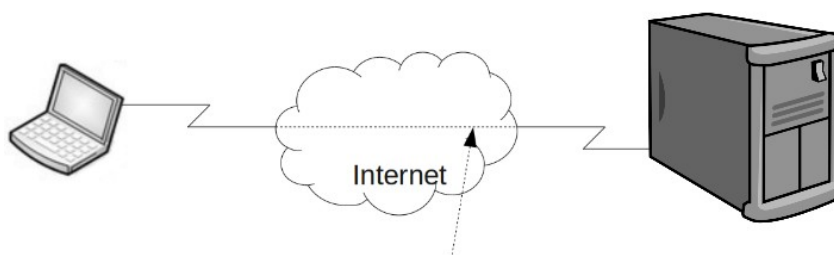
Softwarová konfigurace serveru je dána několika základními požadavky:

- Zabezpečení spolehlivého provozu hostujícího OS .
- Vysoká konfigurovatelnost.
- Zajištění odděleného souběžného provozu VS .

Konfigurace hostujícího OS

- Ubuntu Server 10.04 LTS
- KVM
- Windows XP
 - VLAM Eclipse IDE, integrující SW nástroje VLAM
 - CodeWarrior, verze 5.9.0 (obsahuje i Processor Expert)
 - Vlixicon (VLAM SW pro vzdálené ovládání měřících přístrojů)
 - Agilent Connection Expert, verze 15.5.13009.1
 - Agilent Instrument Explorer, verze 2.1.8910.0
 - Agilent Interactive IO, verze 2.0.8818.2
 - Agilent Measurement Manager, verze 1.8.7.0
 - QdevKit, verze 1.1.8

2.2.3 Zabezpečení vzdáleného přístupu



Bezpečně šifrovaná VPN

Obr. 4: Schéma VPN.

PPTP VPN (Point-to-Point Tunneling Protocol VPN)

Vychází z protokolu PPP (Point-to-Point Protocol). Protokol PPTP přebírá data paketů PPP a dále je zapouzdří do datagramů IP, které nakonec přenesou přes tunel sítě VPN v běžném

Internetu. PPTP podporuje šifrování a kompresi uvedených datových paketů.

Nevýhody protokolu:

- Diskutabilní bezpečnost, záleží na výběru šifrovací techniky, implementace od Microsoftu také není nejvhodnější.

OpenVPN

OpenVPN je open-source projekt podporující mnoho platform (Linux, Solaris, Mac OS X, a další).

S OpenVPN lze:

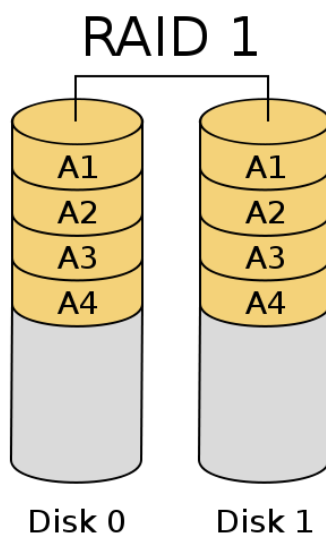
- tunelovat provoz z libovolné IP podsítě přes jediný UDP nebo TCP.
- využívat šifrovací, autentizační a certifikační vlastnosti knihovny OpenSSL k ochraně přenášených dat.
- lze použít šifrování pomocí statického klíče nebo pomocí SSL/TLS certifikátů. [8]

2.3 USB přes IP

Vzhledem k tomu, že některá zařízení používaná v laboratoři nemají ethernet port, ale pouze USB, vyvstal problém jak řešit sdílení těchto zařízení a následnou práci s nimi. Bylo proto nutné najít vhodné řešení USB přes IP. Existuje hardwarová a softwarová varianta tohoto řešení. S ohledem na specifickou používaných zařízení (osciloskop, FITkit.), bylo potřeba provést testy, zda je daný produkt pro určená zařízení vhodný. Jako první na řadu přišla softwarová řešení, která nepotřebovala žádnou investici. K dispozici jsou open-source i komerční řešení, ke kterým však existují trial verze, jenž je možné vyzkoušet zdarma. Vzhledem k vysokým investicím do kvalitního softwarového produktu, byla snaha o nalezení přijatelného hardwarového řešení, k němuž došlo po otestování softwarových variant. Samotné testy s výsledky jsou popsány v praktické části.

2.4 Diskové pole

RAID (Redundant Array of Independent Disks) se používá jako metoda zabezpečení dat proti selhání disku. Zabezpečení je realizováno specifickým ukládáním dat na více nezávislých disků, kdy jsou data uchována i v případě selhání některého z disků. Využíván je RAID1, zrcadlení. Výhodou tohoto řešení je, že k jeho realizaci stačí pouze dva disky. Nevýhodou je ale vyšší redundance, protože stejná data se ukládají na všechny disky v poli. Výhodou je rychlejší čtení, jelikož data se čtou z obou disků najednou, naopak ze stejného důvodu je pomalejší zápis. [9]



Obr. 5: RAID1. [9]

RAID lze realizovat několika způsoby: hardwarový RAID, FakeRAID a softwarový RAID. Hardwarový RAID je finančně náročný, proto nebyl do návrhu řešení zařazen. Softwarový RAID obstarává operační systém. FakeRAID je speciální typ softwarového RAIDU, nespravuje jej však operační systém sám, nýbrž s pomocí BIOSu.

2.4.1 Výhody a nevýhody FakeRAIDu a Linux SW RAIDu

Linux SW RAID

Výhody:

- Pokud server obsahuje hotswap disky a správně nakonfigurovaný systém, lze za plného provozu serveru bez výpadku služeb odstranit poškozený disk a nahradit jej novým. Nastavení nového disku probíhá pomocí příkazové řádky.
- Do Linux SW RAIDu lze zapojit různě velké disky - nezapojují se totiž pouze celé disky, ale libovolná bloková zařízení (diskové oddíly, LVM logické svazky, atp.). Pokud po letech provozu selže jeden ze starých disků a do RAIDu je vložen nový disk s násobnou velikostí, není jeho zbylá volná kapacita promarněna – je možné ji využít na ukládání nekritických dat, popř. lze zbylou kapacitu do RAIDu zapojit později, až se vymění i ten zbývající starý malý disk
- Linux SW RAID je nezávislý na HW počítače. Pokud dojde k havárii HW, lze disky vložit do jakéhokoli jiného počítače a server je okamžitě znovu plně funkční.
- Linux SW RAID je plně funkční, odladěný a stabilní systém, určený pro nasazení v ostrém provozu.

Nevýhody:

- Ve srovnání s plně HW RAIDem často menší rychlost a větší zátěž CPU.

FakeRAID

Výhody:

- Je možný dual-boot Windows-Linux, oba OS vidí RAID jako jeden disk. Vzhledem k tomu, že v laboratoři je provozován server s linuxovým systémem, je tato informace nepodstatná.
- Jednoduché ovládání.

Nevýhody:

- Pokud jeden z disků v RAIDu selže a výrobce FakeRAIDu neposkytuje SW k jeho správě přímo v daném OS, pak je nutné po vložení nového disku restartovat a v BIOSu následně nový disk nastavit. Server bude v té době offline. Tato situace u SW RAIDu nastat nemůže, pokud jsou disky hotswap.
- Pokud se pokazí základní deska, musí se najít nějaká se stejným BIOSem/chipsetem, jinak diskové pole nepůjde znova složit. U RAIDu1 (zrcadlení) půjde provozovat alespoň jeden disk.
- Linux nemusí mít ovladač pro daný BIOS SW RAID, nebo ovladač nemusí být stabilní.
- Většina BIOS SW RAIDů neumožňuje nastavovat parametry typu "Minimální/maximální rychlost synchronizace", což může ovlivnit výkon serveru v době, kdy by se synchronizace prováděla. [10]

2.5 SW/HW serveru VLAM

2.5.1 Vstupní brána projektu (VLAM Gateway)

Jedná se o server založený na OS Linux (Debian 5 Lenny) poskytující následující služby:

- Webový a databázový server (Apache 2 a MySQL 5) hostující webový portál projektu VLAM,
- SW router založený na firewallu Shorewall sloužící pro přesměrování datového spojení mezi virtuálními pracovními stanicemi projektu VLAM a jejich uživateli.

Použitý software:

- Ubuntu Server 10.04 LTS
- MS Windows XP
- KVM
- Apache HTTP Server
- MySQL

Webový portál projektu VLAM poskytuje centralizované přihlašování do systému a přístup ke vzdáleným aplikacím publikovaným prostřednictvím 2X Application Serveru instalovaném na pracovní virtuální stanici, kamerový systém pro vizuální sledování činnosti ovládaného HW zařízení a administrační část sloužící pro detailní správu uživatelů, pracovních stanic a publikovaných aplikací. Úvodní stránka umožňuje vložit přihlašovací údaje a zvolit jednu z aktuálně dostupných/volných virtuálních pracovních stanic, ke které bude uživatel připojen.

Po úspěšném přihlášení uživatele do systému je zobrazena stránka se seznamem vzdálených publikovaných aplikací. Možností je zobrazení vlastního webového rozhraní aplikačního serveru (je požadováno dodatečné přihlášení) zobrazení výstupu kamery sledující ovládané HW zařízení.

Po odhlášení se uživatele z webového portálu VLAM je automaticky zrušeno přesměrování datové komunikace na portálovém serveru a virtuální pracovní stanice je uvolněna dalšímu uživateli.

Webová aplikace VlamGateway obsahuje rovněž administrační část, která umožňuje detailní nastavení systému. Správa webové aplikace VlamGateway lze rozdělit do tří kategorií:

- obecná nastavení aplikace VlamGateway
- správa uživatelů aplikace VlamGateway

- správa pracovních stanic a publikovaných aplikací

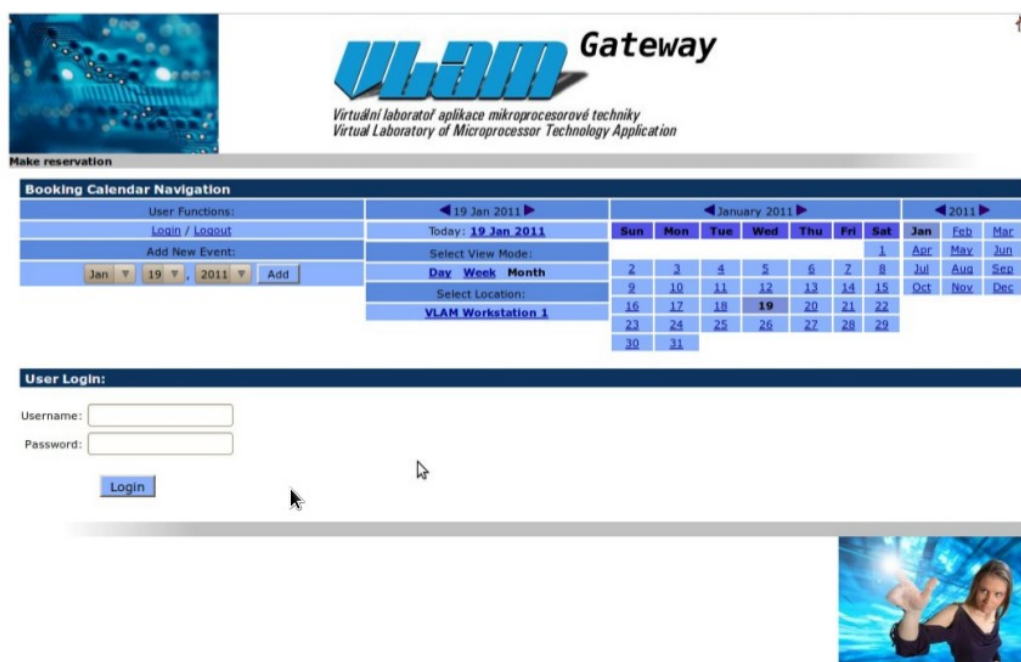
Obecná nastavení aplikace umožňují zvolit čekací doby na přihlášení ke stanici a maximální dobu, po kterou může zůstat na stanici přihlášen jede uživatel. Sekce správy uživatelů aplikace VlamGateway umožňuje:

- definici uživatelů, kteří jsou oprávněni pracovat s aplikací,
- nastavení přístupových práv do jednotlivých částí webové aplikace (práce na virtuální stanici, administrace uživatelských účtů, administrace pracovních stanic a publikovaných aplikací)
- editaci uživatelských účtů a jejich mazání.

Sekce správy pracovních stanic umožňuje:

- vytvářet nové pracovní stanice,
- editovat a mazat existující pracovní stanice,
- publikovat nové aplikace na virtuálních pracovních stanicích,
- editovat a mazat již publikované aplikace,
- nastavení firewallu.

Rezervační systém je dostupný pro všechny registrované uživatele přímo z přihlašovací stránky portálu a umožňuje zarezervovat vybranou aktivní VPS. Rezervaci lze provést jednorázově či opakovaně a to s denní, týdenní, měsíční nebo roční frekvencí. Všechny rezervované termíny jsou přehledně zobrazeny v kalendáři rezervačního systému. [11]



VLAM Gateway
Virtuální laboratoř aplikace mikroprocesorové techniky
Virtual Laboratory of Microprocessor Technology Application

Make reservation

Booking Calendar Navigation

User Functions: Login / Logout Today: 19 Jan 2011

Add New Event: Select View Mode: Day Week Month

Select Location: VLAM Workstation 1

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Jan	Feb	Mar
						1	Apr	May	Jun
2	3	4	5	6	7	8	Jul	Aug	Sep
9	10	11	12	13	14	15	Oct	Nov	Dec
16	17	18	19	20	21	22			
23	24	25	26	27	28	29			
30	31								

User Login:

Username:

Password:

Login

Obr. 6: Vstup do rezervačního systému VLAM Gateway. [11]

2.6 Vlaxicon

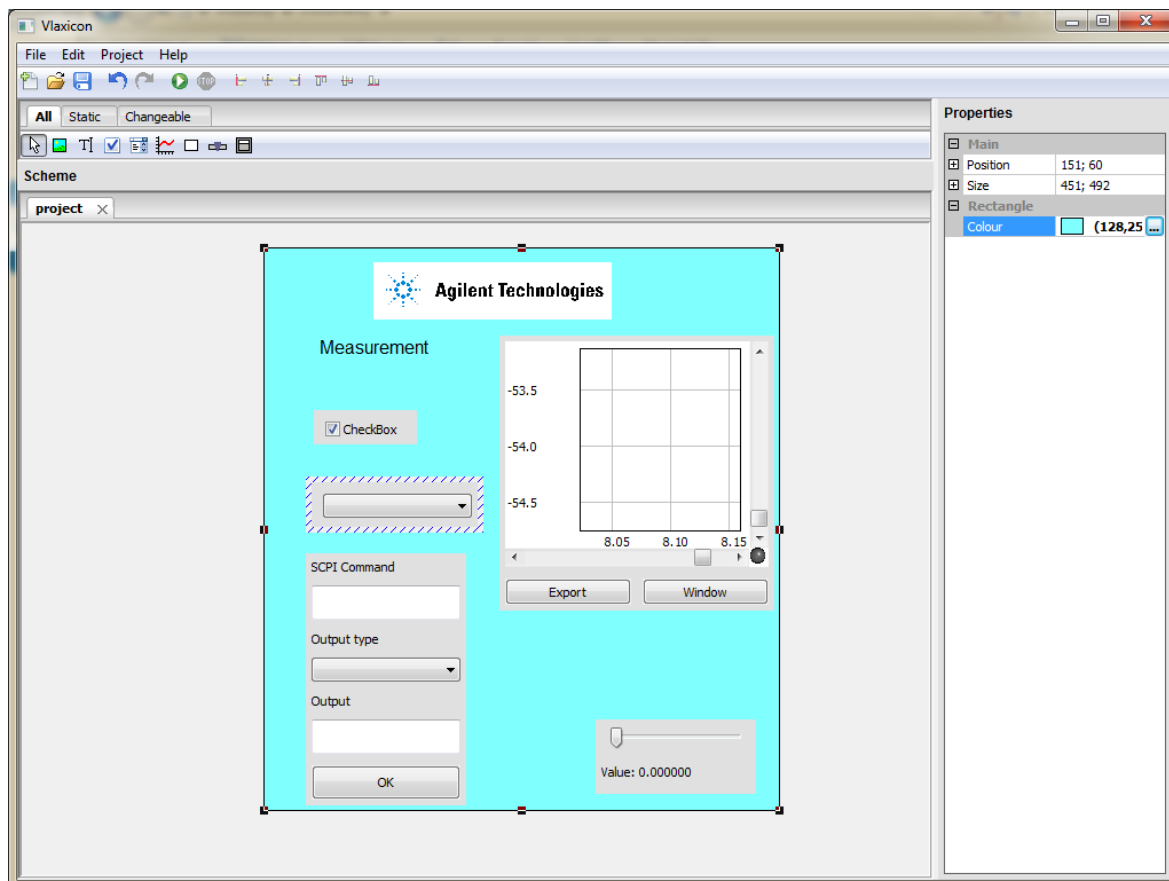
Vlaxicon je program pro vzdálené ovládání měřících přístrojů pomocí PC. Vlaxicon je vyvíjen za použití jazyka C/C++ a knihovny wxWidgets, tím je zajištěna jeho multiplatformnost.

Komunikace pomocí různých protokolů je řešena pomocí pluginů. Pro každou platformu a protokol je vytvořen zvláštní plugin. Zde lze spatřovat velkou výhodu v možnosti budoucího jednoduchého rozšiřování celé aplikace pro komunikaci na jiných systémech a protokolech. To z důvodu, že všechny pluginy jsou vytvořeny na základě společného interface, kdy mají stejné funkce a pouze samotná implementace funkcí je rozdílná kvůli nejednotnému způsobu komunikace na různých platformách a protokolech. V základu programu je vytvořen plugin pro komunikaci na platformě MS Windows a Linux. V případě MS Windows jde o jeden plugin, kdy pomocí ovladače VISA je možné s přístroji komunikovat jak přes USB tak TCP/IP. V případě platformy Linux jsou dva samostatné pluginy pro komunikaci na jednotlivých protokolech USB a VXI (TCP/IP).

Samotné ovládání přístrojů v programu je řešeno pomocí vytvořených prvků, kterým lze nastavovat vlastnosti a pomocí nichž lze nastavovat parametry přístrojů.

Vlaxicon umožňuje:

- komunikovat s přístroji připojenými přes USB,
- komunikovat s přístroji připojenými přes TCP/IP,
- multiplatformní nasazení,
- rozšíření pro další typy připojení a systémy,
- u přístrojů plnit SCPI příkazy,
- uživateli vytvářet projekty pro různé účely a spravovat je,
- manipulaci s přístroji pomocí ovládacích prvků [12]



Obr. 7: Okno programu Vlaxicon s otevřeným projektem. [12]

2.7 Technologie hradlových polí

Programovatelná hradlová pole, známá pod zkratkou FPGA, jsou jedním z nejdůležitějších směrů vývoje integrovaných obvodů s velmi velkou hustotou integrace. Představují populární návrhovou alternativu k aplikačně-specifickým integrovaným obvodům (ASIC), zejména díky nízkým výrobním nákladům z pohledu koncového uživatele, vysokému počtu vyrobených kusů z pohledu výrobce, krátké otočky návrhového cyklu, znovupoužitelnosti v případě FPGA obvodů založených na SRAM technologii, a v poslední době též možnosti dynamicky změnit konfiguraci celého FPGA obvodu nebo jeho částí za běhu. Na druhé straně návrhy používající FPGA technologii poskytují nižší výkon ve smyslu maximální pracovní hodinové frekvence či příkonu než například ASIC obvody a nevyplácí se je používat pro výrobu s více než 10 000 plánovanými kusy.

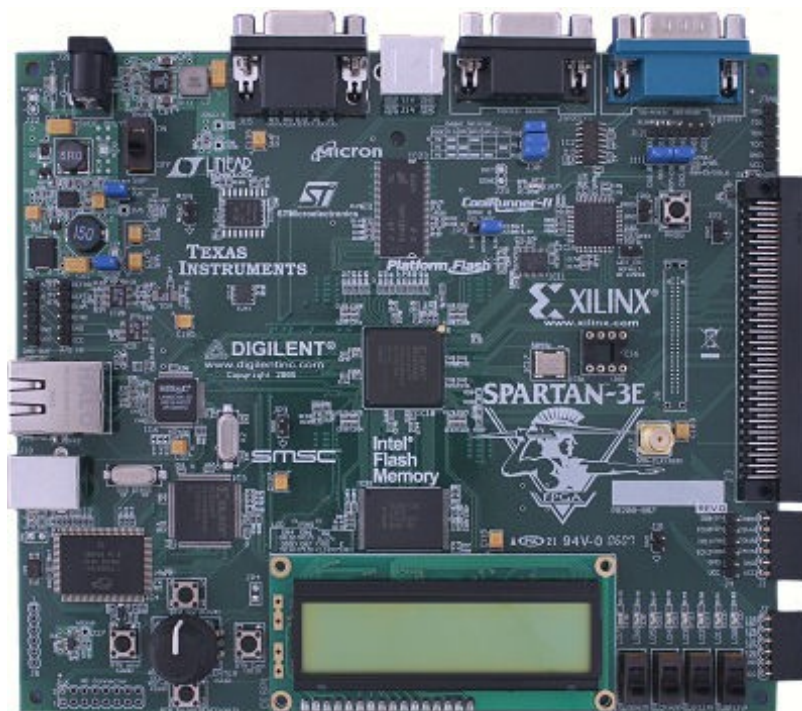
Jeden FPGA obvod lze použít pro implementaci různých uživatelských zapojení, obvodů. Koncept univerzálního předpřipraveného obvodu, který bude vyhovovat mnoha aplikacím, je velmi populární, neboť je to nejefektivnější způsob snižování výrobních nákladů. Existují dva typy takových obvodů: maskou programovatelná hradlová pole (MPGA) a programovatelná hradlová pole (FPGA). Obvody FPGA byly odvozeny přímo z obvodů

MPGA. Prázdný MPGA obvod je standardní čip, který obsahuje jednoduché logické prvky (tranzistory, hradla) a propojovací vodiče určené hlavně pro rozvod napájení umístěné v pravidelném vzoru na ploše čipu. Návrhář implementuje logický obvod v MPGA tvorbou masek kovových vrstev (metalů) v posledním kroku výrobního procesu. To znamená, že koncový uživatel MPGA (návrhář) musí být součástí výrobního procesu, což na něj klade vysoké nároky ve smyslu verifikace návrhu, dodržování návrhových pravidel a dostatečných finančních prostředků. Programovatelná hradlová pole odstraňují tyto problémy tím, že uživatel není součástí výrobního procesu. Cenou, kterou za to platí, je zvýšená složitost nepersonalizovaného obvodu, který musí oproti MPGA navíc obsahovat konfigurační paměť, logiku a programovatelné přepínače. Toto je však bohatě kompenzováno pružností použití FPGA a velkými výrobními sériemi, umožněnými tímto konceptem, z čehož má užitek výrobce i návrhář.

V současnosti se věnují výrobě FPGA obvodů mnoho firem, v projektu bylo využito produktu firmy Xilinx. [13]

2.7.1 Spartan-3E Starter Kit

Deska je poměrně rozšířená a má podporu jak Xilinx tak dalších firem poskytujících návrhové nástroje pro SW i HW. Deska dovoluje realizaci jak malých nízkonákladových řešení s PicoBlaze tak i menších 32bitových systémů s MicroBlaze se středním SW výkonem. [14]



Obr. 8: Xilinx Spartan 3E. [14]

2.7.2 Výukové lekce pro PicoBlaze

V rámci projektu VLAM vyvinula UTIA výukové lekce, které byly integrovány do VLAM prostředí. Jednotlivé lekce lze nalézt na internetových stránkách projektu VLAM: <http://www.vlam.cz/>.

Náplň jednotlivých lekcí:

1. Assembler, C překladač a simulační prostředí.
2. Generování VHDL a implementace systému s procesorem PicoBlaze do FPGA v prostředí Xilinx ISE.
3. Sériová komunikace RS-232 a testování IP jader pomocí procesoru PicoBlaze.
4. Aplikace pro výuku assembleru procesoru PicoBlaze.
5. Popis sadu modulů pro ovládání periferií na vývojové desce Spartan-3E Starter Kit (S3ESK).
6. Zpracování akustických signálů pomocí FPGA (ukázka pokročilého projektu). [15]

2.8 VLAM IDE

Integrované vývojové prostředí bylo vyvíjeno pro začínající programátory embedded systémů v rámci projektu VLAM. Cílem projektu bylo poskytnout uživatelsky přívětivý, intuitivní softwarový nástroj jako podporu hardwarově orientovaných kurzů v rámci výuky.

Důležité části vývoje:

- Návrh HW, který bude realizován v FPGA čipu na vývojové platformě.
- Psaní zdrojového kódu, který bude vykonáván cílovým procesorem.
- Programování vývojové platformy.
- Simulace, debugování a verifikace vytvořeného embedded systému.

Jako vývojové prostředí byl vybrán Eclipse, což je multiplatformní open-source řešení, které se v posledních letech stalo de facto standardem pro vývoj aplikací v nejrůznějších jazycích. [16]

2.8.1 Integrace nástrojů

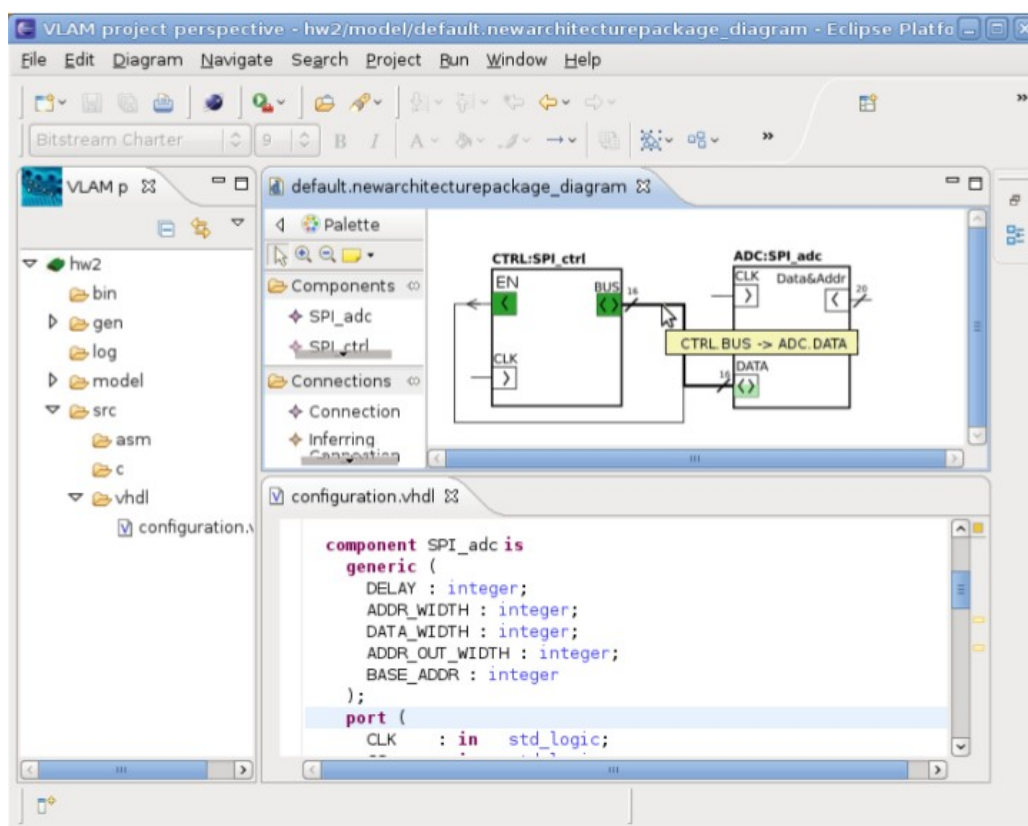
Byl navržen a implementován grafický komponentní editor návrhu hardware, který poskytuje uživatelsky příjemný a intuitivní nástroj pro popis hardwarové části systému cíleného na FPGA. Back-end je vybudován na Eclipse-Modeling Frameworku a front-end na Graphical Editing Frameworku spolu s vlastním designem a implementací.

V rámci projektu byl vyvinut algoritmus pro poloautomatické odvozování propojení hardwarových komponent na syntaktické a částečně i sémantické úrovni. Inferenční

algoritmus má usnadnit uživateli editoru vytváření schémat zapojení. Poté, co uživatel zvolí dvě komponenty k propojení, algoritmus odvodí několik nejlepších propojení (podle ohodnocovací funkce) a uživatel zvolí jedno z navržených řešení. V případě potřeby je možné toto propojení následně podle potřeby ručně upravit. Po ukončení hardwarového návrhu je vygenerován odpovídající zdrojový kód v HDL (Hardware Description Language, VHDL). Tento vygenerovaný kód lze dále manuálně editovat a rozšířit v integrovaném VHDL editoru s využitím Eclipse Verilog Editor plug-inu. Jako další krok může být z VHDL syntetizován výsledný bitstream.

V souladu s kódovací fází může uživatel psát zdrojový kód pro procesory cílové platformy s použitím integrovaného editoru pro jazyk symbolických adres a jazyk C. Výsledný zdrojový kód je kompilován integrovanými kompilátory.

Finální krok zahrnuje proces kompilace zdrojového kódu a popis hardware syntézy. Využívá se prostředí příkazové řádky nástroje QdevKit, který generuje odpovídající make soubory obsahující všechny potřebné cesty ke zdrojovým kódům, cílové architektuře. Dále s použitím GNU make utility lze kompilovat a slučovat všechny potřebné zdrojové kódy do projektu. [16]



Obr. 9: VLAM IDE.[16]

2.9 PicoBlaze C Compiler (PBCC)

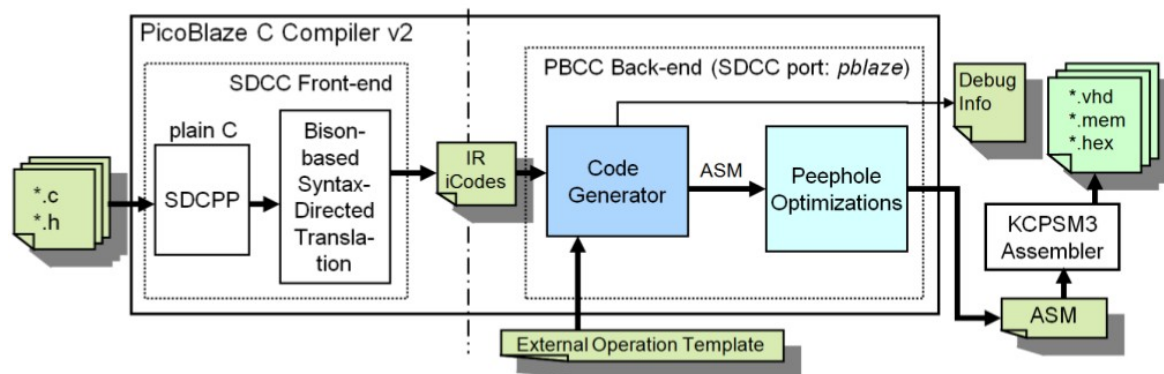
2.9.1 Historie

Firma Xilinx, která patří na světě k předním výrobcům FPGA a nástrojům pro vývoj na této platformě, uveřejnila již před mnoha lety velmi jednoduchý osmibitový mikrokontrolér jménem PicoBlaze, u něhož je dnes dostupná už jeho třetí verze. Pro usnadnění psaní řídicího kódu pro tento soft-core mikrokontrolér bylo rozhodnuto vytvořit překladač podmnožiny jazyka C.

Již původní implementace PicoBlaze C Compiler byla postavena na rámci SDCC (Small Device C Compiler), ale zadní část tvořil kód ve Squeak Smalltalku. V té době totiž nebylo žádoucí psát zadní část v jazyce C. Bohužel pro přijetí kompilátoru komunitou bylo třeba mít překladač o řád lepší než ten původní a především implementovaný v jazyce lépe konzistentním s SDCC a to v jazyce C. Proto se pracuje na portaci a rozšíření překladače PBCC na novou verzi PBCCv2.

2.9.2 Architektura

Vysokoúrovňový návrh PBCC je zobrazen níže na obrázku. Přední část SDCC začíná SDCCPP (adaptovaný GCC preprocesor) a GNU Bison, které provádí překlad řízený syntaxí ze zdrojového kódu v C do vnitřního kódu (skládá se z tříadresného kódu s přídatnými informacemi vyšší úrovně jako např. datové typy). Protože v SDCC neexistuje žádné spolehlivé rozhraní pro vlastní implementaci zadní části překladače (tzv. port) a dokumentace je nedostačující, bylo provedeno několik experimentů za účelem prozkoumání vnitřního kódu SDCC a zdokumentování způsobu vytvoření nového portu. Ve výsledku je PicoBlaze C Compiler složen z několika zřetězených command-line utilit. [17]



Obr. 10: Architektura PicoBlaze C Compilera (PBCC). [17]

II. PRAKTICKÁ ČÁST

3 Virtualizace 8 Windows XP hostů na Linuxovém serveru

Cílem je otestovat, který virtualizační nástroj nejlépe vyhovuje pro virtualizaci 8 Windows XP hostů na Linuxovém serveru. Tedy které řešení je nejvhodnější pro zadaný problém. K tomu posloužila série zátěžových testů, běžících na všech 8 vytvořených MS Windows XP hostech. Dále v rozhodování hrála roli samozřejmě cena, uživatelská přívětivost, která je důležitá zejména při budoucí údržbě.

3.1 Použitý hardware

- Server: IBM System x3400
- HP ProBook 4510s

3.2 Výběr virtualizačních nástrojů

- Xen [18]
- KVM [19]
- VMware server 2.0.2 [20]
- VMware ESX 3.5 [20]

3.3 Výběr distribuce Linuxu

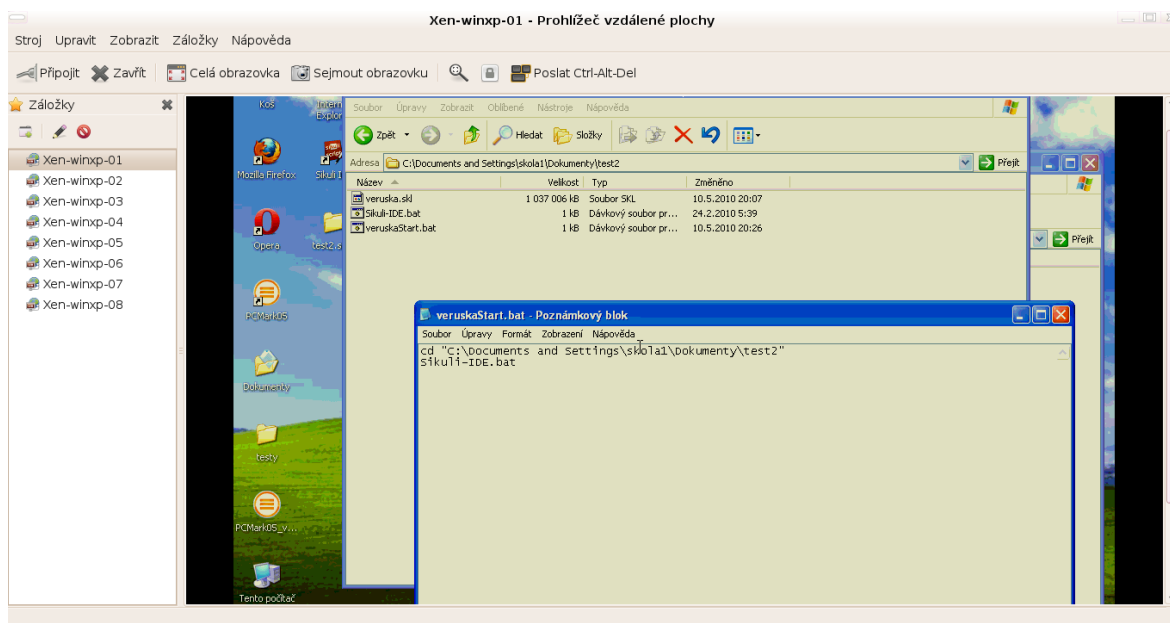
- Debian Lenny (5.0.2) [21]
- Fedora 13 [22]
- Arch Linux

3.4 Přístup k virtuálním strojům

Vinagre umožňuje, aby si uživatel mohl prohlížet (nebo i ovládat) vzdálené pracovní prostředí. K tomu se připojuje pomocí protokolu VNC.

Vinagre umí automaticky najít všechna sezení, která vzdálené přihlášení umožňují, takže je připojení téměř bezpracné. Toto je zajištěno díky technologii Avahi.

V jednu chvíli je možné si otevřít libovolný počet sezení; Vinagre je otevře v kartách. Oblíbená vzdálená prostředí je navíc možné uložit jako záložky. [23]



Obr. 11: Prostředí programu Vinagre.

3.5 Výběr zátěžových testů

Benchmark, který se automaticky spouští pomocí SIKULI [24] ze začátku vypadal jako jednoduché řešení, avšak ve finále se ukázalo nestabilní, problém byl zejména v dokončení testů na 8 virtuálních strojích. Ve většině případů alespoň na jednom stroji test selhal a nedoběhl do konce, proto se muselo od těchto testů upustit a přejít k programování testovacích skriptů.

3.6 Virtualizační nástroje – základní konfigurace

3.6.1 Xen

Postup:

Instalace Xenu:

```
# apt-get install xen-hypervisor-3.2-1-amd64 xen-linux-system-2.6.26-1-  
xen-amd64 xen-utils-3.2-1 xenstore-utils xenwatch xen-shell xen-tools
```

Úprava souboru /etc/modules:

```
...  
loop max_loop=64
```

Úprava souboru /etc/xen/xend-config.sxp:

```
...  
# zakomentovat  
#(network-script network-dummy)  
...  
# odkomentovat  
(network-script network-bridge)  
(vif-script vif-bridge)  
...
```

Restart serveru:

```
# reboot
```

Vytvoření LVM logického svazku pro Windows ve skupině svazků „fs“:

```
# lvcreate -L 20G -n winxp-01 fs
```

Úprava souboru /etc/xen-tools/xen-tools.conf

```
...  
#dir = /home/xen  
# If you don't wish to use loopback images then you may specify an  
# LVM volume group here instead  
lvm = fs  
...
```

Vytvoření ISO obrazu instalačního CD MS Windows XP:

```
$ dd if=/dev/cdrom of=/home/veruska/winxp.iso bs=256M
```

Vytvoření konfiguračního souboru virtuálního stroje /etc/xen/winxp-01.cfg:

```
...
# Initial memory allocation (in megabytes) for the new domain.
#
# WARNING: Creating a domain with insufficient memory may cause out of
#          memory errors. The domain needs enough memory to boot kernel
#          and modules. Allocating less than 32MBs is not recommended.
memory = 448
...
# A name for your domain. All domains must have different names.
name = "winxp-01"
...

# Optionally define mac and/or bridge for the network interfaces.
# Random MACs are assigned if not given.
# type=ioemu specify the NIC is an ioemu device not netfront
vif = [ 'type=ioemu,bridge=br0' ]
#vfb = [ 'type=vnc' ]
#-----
-----
# Define the disk devices you want the domain to have access to, and
# what you want them accessible as.
# Each disk entry is of the form phy:UNAME,DEV,MODE
# where UNAME is the device, DEV is the device name the domain will see,
# and MODE is r for read-only, w for read-write.
disk = [ 'phy:/dev/fs/winxp-01,ioemu:hda,w',
         'file:/home/veruska/winxp.iso,ioemu:hdc:cdrom,r' ]
...
# boot on floppy (a), hard disk (c), Network (n) or CD-ROM (d)
# default: hard disk, cd-rom, floppy
boot="d"
...
# address that should be listened on for the VNC server if vnc is set.
# default is to use 'vnc-listen' setting from /etc/xen/xend-config.sxp
vnclisten="192.168.1.50"

# set VNC display number, default = domid
vncdisplay=1
...
```

```
# set the real time clock to local time [default=0 i.e. set to utc]
localtime=1
...
```

Nastavení sítě v `/etc/network/interfaces`:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug eth0
auto eth0
#dhcp zmeneno na manual
iface eth0 inet manual
up sysctl -w net.ipv4.conf.eth0.proxy_arp=1

auto br0
iface br0 inet static
up sysctl -w net.ipv4.conf.br0.proxy_arp=1
address 192.168.1.50
netmask 255.255.255.0
bridge_ports eth0
bridge_fd 0
bridge_stp 0
```

3.6.2 KVM

Zjištění podpory virtualizace: [27]

```
$ egrep '(vmx|svm)' /proc/cpuinfo
```

Postup:

Instalace potřebných balíčků:

```
# apt-get install virt-manager libvirt-bin kvm
```

Nastavení sítě v /etc/network/interfaces:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug eth0
auto eth0
iface eth0 inet manual
up sysctl -w net.ipv4.conf.eth0.proxy_arp=1

auto br0
iface br0 inet static
up sysctl -w net.ipv4.conf.br0.proxy_arp=1
address 192.168.1.50
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
bridge_ports eth0
bridge_fd 0
bridge_stp 0
```

3.6.3 VMware Server

VMware Server se nepodařilo ani nainstalovat, proto byla zavrženo jeho nasazení hned v počátku.

3.6.4 VMware ESX Server

Instalace VMware ESX Server je jednoduchá, avšak celková uživatelská přívětivost bez použití velmi drahého SW vCenter je oproti Xenu a KVM horší. Nepodařilo se vytvořit snapshoty, proto by tvorba virtuálních strojů byla nepohodlná a zdlouhavá.

3.7 Tvorba virtuálních strojů

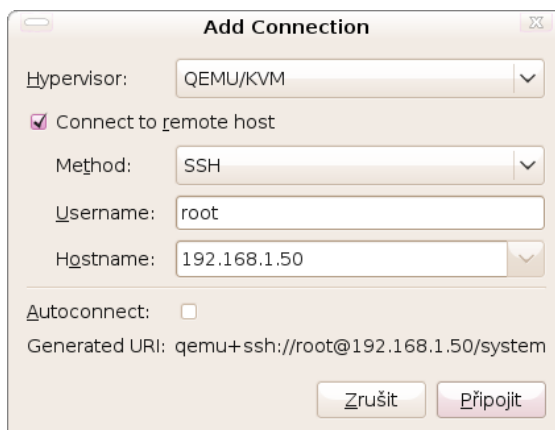
Základním předpokladem bylo vytvořit jeden originální virtuální stroj, který se bude klonovat na celkových osm. Výhodou tohoto řešení je, že stačilo nainstalovat a nastavit jeden virtuální stroj, který se pomocí snapshotů dále naklonoval a nastavení všech osmi virtuálních strojů tedy odpovídalo originálnímu.

3.7.1 Správa virtuálních strojů

Spravovat virtuální stroje je možné buď z konzole, nebo pomocí programu Virtual machine-manager. [28]

Práce s Virtual machine-managerem:

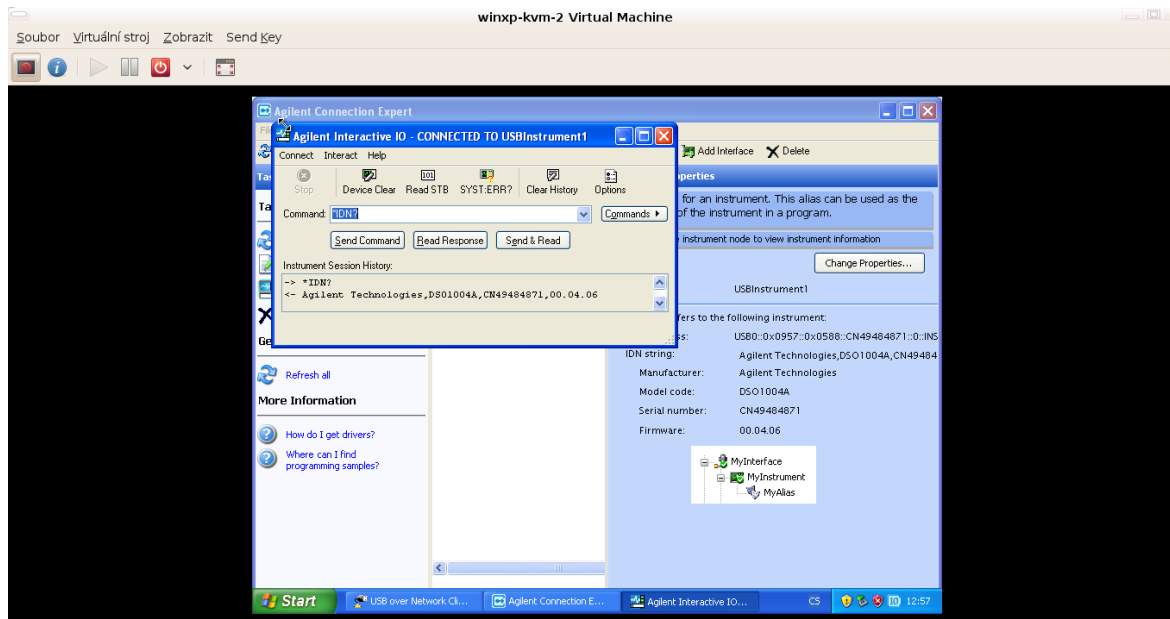
Soubor → Přidat připojení



Obr. 13: Virt-manager, přidání nového připojení.



Obr. 14: Ukázka virtuálních strojů.



Obr. 15: Ukázka práce ve virtuálním stroji pomocí programu Virt-Manager.

SSH:

Vygenerování veřejného klíče:

```
$ ssh-keygen -t rsa
```

Zkopírování veřejného klíče na server:

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.1.50
```

Připojení:

```
$ ssh root@192.168.1.50
```

Práce se snapshoty:

Tvorba snapshotu:

```
# lvcreate --size 5G --snapshot --name winxp-01 /dev/fs/winxp-original
```

Smazání snapshotů

```
# for i in `seq 1 8`; do lvremove -f /dev/fs/winxp-0${i}; done
```

Kontrola:

```
#lvs
  LV          VG   Attr   LSize   Origin              Snap%   Move Log Copy%
Convert
  root        fs   -wi-ao 9.31G
  winxp-01    fs   swi-a- 5.00G  winxp-original      0.00
  winxp-original fs   owi-a- 20.00G
```

Práce s virtuálními stroji - Xen*Zkopírování konfiguračních souborů*

```
# for i in `seq 1 8`; do cp winxp-01.cfg winxp-0${i}.cfg; done
```

Nastartování virtuálních strojů

```
# for i in `seq 1 8`; do xm create winxp-0${i}.cfg; done
```

Vypnutí virtuálních strojů

```
# for i in `seq 1 8`; do xm destroy winxp-0${i}; done
```

Práce s virtuálními stroji – KVM*Nastartování virtuálních strojů*

```
# for i in `seq 1 8`; do virsh start winxp-0${i}; done
```

Vypnutí virtuálních strojů

```
# for i in `seq 1 8`; do virsh shutdown winxp-0${i}; done
```

Paravirtualizace

Podpora paravirtualizace pro Windows XP hosty je jen velmi omezená, ve formě ovladačů pro disková a síťová zařízení. Bohužel tyto ovladače jsou velmi nestabilní, takže se nehodí pro produkční nasazení, a proto k samotnému testování s paravirtualizací nedošlo.

3.8 Výsledky testů: KVM versus XEN

3.8.1 Pro jeden virtuální stroj

Tab. 3: Testování jednoho virtuálního stroje: KVM versus XEN.

Na disk minutu zapisují data a zjišťují,	
kolik bylo zapsáno kB bloků (velikost 100 MB)- se snapshotem a bez:	
XEN:	16963
	16991
KVM	16997
	16994

3.8.2 Pro osm virtuálních strojů

Tab. 4: Testování 8 virtuálních strojů - XEN, doba zápisu 100 MB.

XEN: Na disk zapisují náhodná data po kBytových blocích, velikost souboru 100MB:											
Číslo měření	1	2	3	4	5	6	7	8	9	10	Průměr:
Jednotky	[s]	[s]	[s]	[s]	[s]	[s]	[s]	[s]	[s]	[s]	[s]
Číslo stroje											
1	394,53	398,09	396,14	389,14	389,75	391,61	390,25	396,250	394,98	392,03	353,65
2	395,94	399,27	396,13	391,16	389,84	393,75	390,13	401,03	391,72	395,67	394,46
3	396,31	396,41	398,36	389,50	390,41	391,25	389,59	400,30	397,59	394,59	394,43
4	397,13	396,14	397,53	390,59	390,38	389,58	389,28	400,14	396,67	395,17	394,26
5	393,36	398,83	397,23	389,25	389,52	392,09	390,00	397,00	396,44	394,53	393,82
6	400,50	397,45	396,11	391,39	391,17	392,52	389,11	397,70	395,30	395,45	394,67
7	392,44	397,44	399,84	390,27	390,78	392,19	390,25	400,36	394,13	395,44	394,31
8	392,59	397,05	397,84	390,67	390,67	391,70	389,81	399,50	395,41	392,59	393,78
Průměr:	389,17										

Tab. 5: Testování 8 virtuálních strojů - KVM, doba zápisu 100 MB.

KVM: Na disk zapisují náhodná data po kBytových blocích,						
velikost souboru 100MB:						
Číslo měření	1	2	3	4	5	Průměr:
Jednotky	[s]	[s]	[s]	[s]	[s]	[s]
Číslo stroje						
1	382,27	375,41	381,02	373,47	378,33	378,10
2	380,77	373,33	375,63	370,16	382,52	376,48
3	386,42	369,22	374,89	372,47	369,52	374,50
4	380,06	375,66	376,47	371,16	373,08	375,28
5	381,08	372,72	368,30	372,17	369,64	372,78
6	387,41	374,61	380,88	374,81	372,69	378,08
7	381,64	380,02	366,89	378,34	372,81	375,94
8	384,94	374,63	375,06	380,39	377,05	378,41
Průměr:	376,2					

Tab. 6: Testování 8 virtuálních strojů - XEN, kB/min.

XEN: Na disk minutu zapisují data a zjišťují, kolik bylo zapsáno kB bloků:											
Číslo měření	1	2	3	4	5	6	7	8	9	10	Průměr:
Jednotky	[kB]	[kB]	[kB]	[kB]	[kB]	[kB]	[kB]	[kB]	[kB]	[kB]	[kB]
Číslo stroje											
1	15990	16162	15625	15642	15672	16695	16236	15870	15949	15629	15947,0
2	15866	16085	15698	15800	15673	16635	16167	15694	15865	15940	15942,3
3	15967	16116	15841	15706	15600	16955	16174	15711	15686	15788	15954,4
4	15931	16136	15797	15780	15552	16642	16078	15796	15906	16027	15964,5
5	15920	16179	15677	15715	15739	16572	16332	15725	15912	15974	15974,5
6	15848	16090	15718	15726	15696	16650	16297	15849	15866	15896	15963,6
7	15920	16182	15706	15885	15701	16572	16182	15913	15884	15779	15972,4
8	15913	16272	15779	15679	15709	16675	16212	15735	15806	15842	15962,2
Průměr:	15960,11										

3.8.3 Subjektivní porovnání z pohledu uživatele

Tab. 8: Výsledky testů virtualizace - subjektivní porovnání z pohledu uživatele.

	Instalace	Snapshoty	Uživatelská přívětivost
XEN			
KVM			
Vmware Server			
Vmware ESX			

3.9 Shrnutí výsledků

Výsledkem testování virtualizačního nástroje bylo nalezení nejvhodnějšího řešení pro náš záměr. Jako vítěz vyšlo z těchto testů KVM. Rozhodnutí však nebylo jednoduché, jelikož souboj mezi Xenem, KVM a VMware ESX byl celkem vyrovnaný. Výsledky testů Xen a KVM byly téměř stejné, avšak KVM vykazovalo při samotné práci v hostovaném systému rychlejší odezvu oproti Xenu. Dalším plusem pro KVM je jednoduchá instalace a uživatelsky příjemnější ovládání. V obou systémech funguje bez problémů tvorba diskových snapshotů, která ulehčuje správu virtuálních strojů. Oba systémy se při testech chovali stabilně. U VMware přicházelo v úvahu nasazení pouze produktu VMware ESX Server, jelikož VMware Server, který je zdarma, vykazoval nestabilitu už při samotné instalaci. Instalace VMware ESX Server byla bez problémů, testy pro jeden virtuální stroj dopadly o dost lépe jako u Xenu a KVM, avšak bez použití produktu vCenter je ovládání nepřívětivé a nastal také problém s diskovými snapshoty, které nešly vytvořit. Produkt vCenter nebyl přijatelný díky vysoké pořizovací ceně. Dále byly neúspěšně testovány paravirtualizační ovladače pro hostované systémy. Ovladače, které se podařilo nainstalovat

způsobovaly BSOD a zbylé nešlo nainstalovat vůbec. Při testování byla snaha o zprovoznění jednotlivých řešení, proto nebyl použit například jen jeden testovací operační systém, ale byla například vyzkoušena i Fedora, ve které bylo nejaktuálnější KVM a Virt-manager. Prioritou číslo jedna byla samozřejmě stabilita systému, proto bylo nutné od některých nabízených řešení upustit, například od paravirtualizace, která se jevila jako velmi nestabilní.

4 Realizace sdílených diskových polí

Ke správnému provozování serveru bezesporu patří realizace záloh s možností co nejjednodušší výměny vadného disku. K tomuto účelu poslouží RAID1 a také funkce hotplug. Obě řešení je potřeba řádně vyzkoušet na živém serveru. Vzhledem k tomu, že vybraný hardware nabízí FakeRAID, je také potřeba porovnat obě řešení a jejich výkon.

4.1 Použitý hardware

- FUJITSU RAID 0/1 SAS based on LSI

4.2 Postup testování

Cílem testování je ověření funkčnosti RAIDu 1. Dále se porovnává FakeRAID a softwarový RAID a ověřuje se funkce hot-plug v obou provedeních RAIDu. K testování výkonu byl použit benchmark Bonnie++.

4.3 Bonnie++

Bonnie++ testuje:

- rychlost čtení a zápisu dat.
- počet seeků, které mohou být vykonány za sekundu.
- počet operací nad metadaty za sekundu. [29]

Instalace:

```
# apt-get install bonnie++
```

```
# mkdir -m 0777 /var/bonnie
```

Spouštění testu pod root:

```
# bonnie++ -d bonnie -u root:root -n 16:10000:16:64
```

Ukládání výstupu do textového souboru:

```
# echo  
"1.96,1.96,dominion,1,1304190533,6544M,,109,98,67246,51,23772,11,907,98,1  
14385,14,204.7,8,16,10000,16,,64,2201,50,30767,96,9684,52,4166,42,30360,9  
9,10610,65,193ms,2159ms,700ms,23894us,26586us,741ms,60304us,378us,585us,3  
51us,293us,355us" >> bonnie2.txt
```

Generování výsledné tabulky:

```
# cat bonnie2.txt | bon_csv2html > bonnie2h.html
```

Kopírování výsledků ze serveru pro následné zpracování:

```
# scp root@192.168.1.105:/var/bonnie3h.html ./
```

4.4 FakeRAID

FakeRAID se vytváří přímo v BIOSu karty.

4.5 Softwarový RAID

Softwarový RAID se vytváří při instalaci linuxové distribuce. K testování byl vybrán Debian 5, jelikož je momentálně nasazen na VLAM serveru.

Postup:

- Vytvořit na prvním disku primární oddíl o velikost 200 MB, přípojný bod: /boot, ext2.
- Vytvořit na druhém disku primární oddíl o velikosti 200 MB, ve volbě dalšího nastavení vybrat nepoužívat, jelikož se při zavádění OS použije /boot oddíl na prvním disku.
- Ze zbývajících místa na jednotlivých discích vytvořit oddíly pro použití v RAIDu.
- Nakonfigurovat RAID: použít fyzické zařízení pro RAID → konfigurování softwarového RAIDu → vytvoření MD zařízení → RAID1 → počet aktivních zařízení pro RAID1 pole → použít jako ext3 → přípojný bod /.

4.6 Hot-plug

Při testování hotplug je potřeba ověřit funkčnost výměny disku za chodu serveru.

Postup – FakeRAID:

- Stahování velkého souboru, aby byla zajištěna vytíženost disku, např:

```
# wget
http://download.fedoraproject.org/pub/fedora/linux/releases/test/15
-Beta/Fedora/i386/iso/Fedora-15-Beta-i386-DVD.iso
```

- Vytažení jednoho ze dvou disků.
- Kontrola stavu, stahování dat by mělo probíhat bez problémů.
- Návrat disku zpět do serveru.
- Opětovná kontrola stavu stahování dat.

Postup – Softwarový RAID

- Kontrola pole [30]:

```
# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Sun May  1 18:00:54 2011
    Raid Level : raid1
    Array Size : 244001656 (232.70 GiB 249.86 GB)
  Used Dev Size : 244001656 (232.70 GiB 249.86 GB)
    Raid Devices : 2
  Total Devices : 2
    Persistence : Superblock is persistent
    Update Time : Sat May 14 18:16:45 2011
    State : clean

Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0

    Name : dominion:0 (local to host dominion)
    UUID : f101b9c7:629f9da2:ef75e998:faffe8aa
    Events : 950

   Number   Major   Minor   RaidDevice State
     0         8       2         0     active sync   /dev/sda2
     1         8      18         1     active sync   /dev/sdb2
```

- U softwarového RAIDu je první potřeba označit disk jako vadný:

```
# mdadm --fail /dev/md0 /dev/sdb2
mdadm: set /dev/sdb2 faulty in /dev/md0
# mdadm --detail /dev/md0
...
    State : clean, degraded
  Active Devices : 1
Working Devices : 1
  Failed Devices : 1
Spare Devices : 0

   Number   Major   Minor   RaidDevice State
     0         8       2         0     active sync   /dev/sda2
     1         0       0         1     removed
     1         8      18         -     faulty spare   /dev/sdb2
```

- Odstranění disku:

```
# mdadm --remove /dev/md0 /dev/sdb2
mdadm: hot removed /dev/sdb2 from /dev/md0
# mdadm --detail /dev/md0
...
  Raid Devices : 2
  Total Devices : 1
    State : clean, degraded
  Active Devices : 1
  Working Devices : 1
  Failed Devices : 0
  Spare Devices : 0

   Number   Major   Minor   RaidDevice State
     0         8         2         0     active sync   /dev/sda2
     1         0         0         1     removed
```

- Přidání nového disku (fyzicky).
- Přidání blokového zařízení do RAIDu:

```
# mdadm --add /dev/md0 /dev/sdb2
mdadm: re-added /dev/sdb2
# mdadm --detail /dev/md0
...
  Raid Devices : 2
  Total Devices : 2
    State : clean, degraded, recovering
  Active Devices : 1
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 1
  Rebuild Status : 5% complete

   Number   Major   Minor   RaidDevice State
     0         8         2         0     active sync   /dev/sda2
     1         8        18         1     spare rebuilding /dev/sdb2
```

Výsledky:

Při FakeRAIDu funguje hotplug bez problémů. Zapisování dat zůstalo nepřerušeno. Manipulace s diskem server zvládl bez zaváhání. U softwarového RAIDu také nedošlo k problémům.

4.7 Výsledky testování

Tab. 9: Výsledky testů - Bonnie++ (Sequential Output, Input, Random Seeks).

Version 1.96		Sequential Output						Sequential Input				Random Seeks	
	Size	Per Char		Block		Rewrite		Per Char		Block		/sec	% CPU
		K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU		
fake RAID													
1	6544M	112	99	67280	51	24143	11	868	97	113847	15	206.6	8
	Latency	113ms		2328ms		2718ms		57845us		26562us		758ms	
2	6544M	110	99	66788	51	23499	11	913	98	114300	14	206.7	8
	Latency	111ms		2380ms		520ms		26763us		27788us		3083ms	
3	6544M	111	99	66844	50	23818	11	840	97	114292	15	221.9	4
	Latency	108ms		2397ms		520ms		65326us		27728us		797ms	
4	6544M	111	99	65777	49	23823	11	836	98	113664	14	205.1	8
	Latency	113ms		2374ms		553ms		49823us		117ms		743ms	
5	6544M	109	98	67246	51	23772	11	907	98	114385	14	204.7	8
	Latency	193ms		2159ms		700ms		23894us		26586us		741ms	
Softwarový RAID													
1	6544M	109	99	61149	49	34954	20	924	99	103927	16	383.0	17
	Latency	128ms		5275ms		7152ms		20920us		35209us		160ms	
2	6544M	109	99	61801	48	34240	19	908	99	102538	16	380.4	17
	Latency	120ms		4064ms		5979ms		30333us		42224us		438ms	
3	6544M	110	99	63767	50	35670	21	820	96	100961	16	376.1	17
	Latency	104ms		4520ms		4032ms		67030us		43716us		201ms	
4	6544M	110	99	68857	54	35673	19	888	98	106301	17	389.7	10
	Latency	91304us		2428ms		722ms		46545us		33027us		782ms	
5	6544M	110	99	68636	54	35462	19	874	98	108437	17	355.6	16
	Latency	99436us		721ms		518ms		29979us		33987us		800ms	

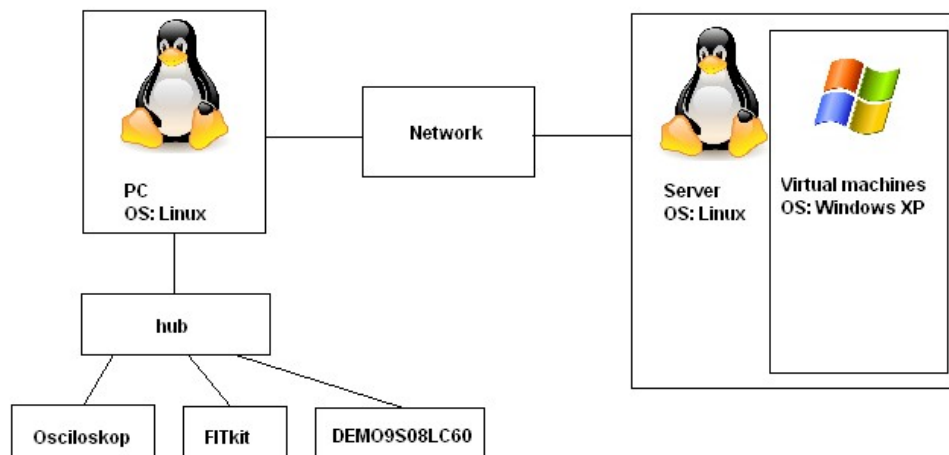
Tab. 10: Výsledky testů - Bonnie++ (Sequential Create, Random Create).

Version 1.96						Sequential Create						Random Create					
Size	Num Files	Max Size	Min Size	Chunk Size	Create		Read		Delete		Create		Read		Delete		
					/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	
fake RAID																	
1	6544M	16	10000	16	64	2324	51	30756	96	9731	52	4063	42	30388	99	11186	68
	Latency				Latency	64424us	303us		584us		323us		392us		422us		
2	6544M	16	10000	16	64	2252	50	30715	95	10015	54	4166	42	30325	99	10839	66
	Latency				Latency	72882us	296us		597us		396us		354us		286us		
3	6544M	16	10000	16	64	2111	47	30550	95	9990	53	4122	42	30264	99	11014	67
	Latency				Latency	46875us	436us		606us		338us		427us		297us		
4	6544M	16	10000	16	64	2297	51	30855	96	9630	51	4035	41	30116	99	11012	68
	Latency				Latency	60078us	437us		573us		397us		385us		295us		
5	6544M	16	10000	16	64	2201	50	30767	96	9684	52	4166	42	30360	99	10610	65
	Latency				Latency	60304us	378us		585us		351us		293us		355us		
Softwarový RAID																	
1	6544M	16	10000	16	64	2590	59	31505	98	14828	82	4882	50	30426	98	13783	86
	Latency				Latency	1470ms	384us		598us		420us		395us		386us		
2	6544M	16	10000	16	64	2680	61	30623	95	14461	82	4839	51	30386	98	13126	87
	Latency				Latency	713ms	422us		606us		397us		391us		482us		
3	6544M	16	10000	16	64	2558	60	30210	94	14709	82	4806	50	30102	99	12881	85
	Latency				Latency	1464ms	396us		571us		313us		305us		416us		
4	6544M	16	10000	16	64	2761	63	30284	95	15694	86	4868	50	30232	98	13881	86
	Latency				Latency	152ms	295us		594us		333us		298us		293us		
5	6544M	16	10000	16	64	2633	60	30064	94	15239	84	4779	50	30318	99	13840	86
	Latency				Latency	163ms	396us		444us		358us		385us		344us		

Jak lze vyčíst z výše uvedených tabulek, výsledky jsou poměrně vyrovnané.

5 USB over IP: Softwarové řešení

Cílem bylo zajistit pomocí vhodného softwaru přístup k zařízením z virtuálních strojů a jejich následné ovládání. Návrh:



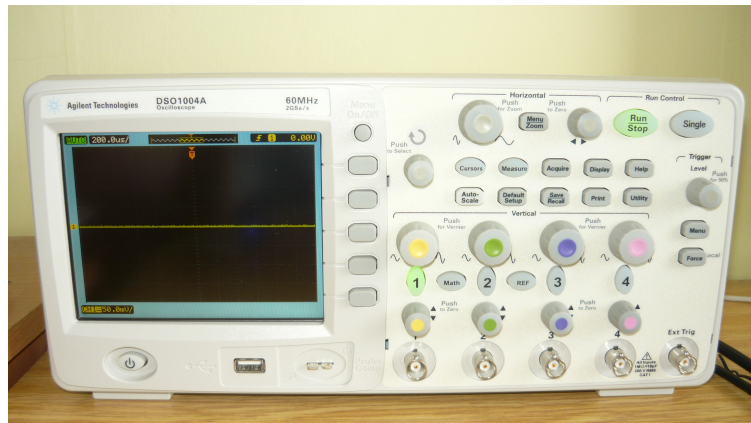
Obr. 16: Návrh USB over IP.

Vybraná řešení obsahují serverovou a klientskou část. Serverová část se instaluje na počítač, kde jsou připojena jednotlivá zařízení, klientská na počítač, ze kterého se přistupuje k USB zařízením. Testování a vybírání nejlepšího produktu probíhalo v několika fázích:

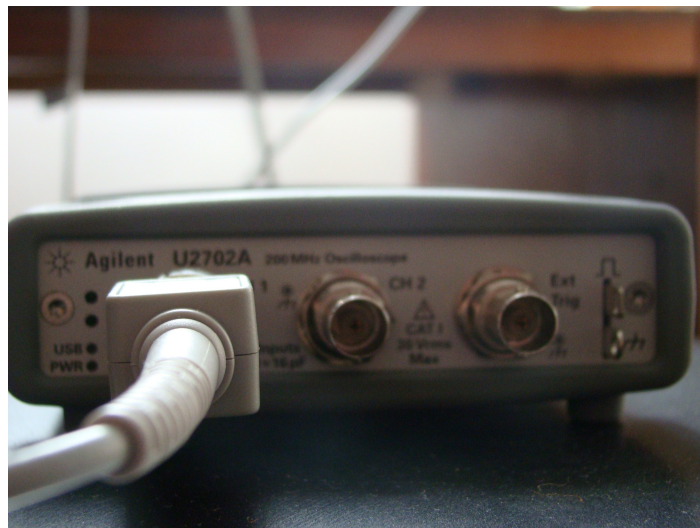
1. Příprava vhodného testovacího prostředí: Instalace operačních systémů na potřebné počítače, instalace ovládacího softwaru pro jednotlivá zařízení.
2. Instalace produktů pro řešení USB over IP problému.
3. Kontrola spojení klient-server:
 - připojené jedno zařízení – kontrola správné činnosti na klientském počítači. Jako první se kontroluje flash disk, při jeho správné funkčnosti se dále pokračuje s potřebnými USB zařízeními a dochází ke kontrole jejich funkčnosti spuštěním vzorových programů či zadáváním základních ovládacích příkazů.
 - Prostřednictvím hubu je připojeno více zařízení, na které se přistupuje nejprve z jednoho a následně z několika klientských virtuálních strojů. Kontroluje se přístupnost na neobsazená zařízení a nemožnost použití již obsazeného zařízení jiným klientským strojem.
4. Kontrola schopnosti pracovat správně s připojenými zařízeními pomocí softwaru doporučeného k ovládání jednotlivých zařízení.
5. Vyhodnocení testů.

5.1 Použitý hardware

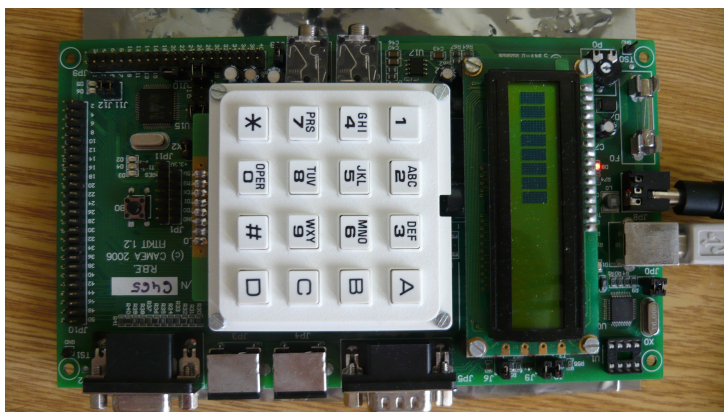
- IBM System x3400
- HP ProBook 4510
- Stolní PC:
 - Procesor: AMD Athlon64 3000+
 - RAM: 2048 MB
 - HDD: 200+640+40GB
 - Graf. karta: Gigabyte GeForce 6600GT 128MB
 - Zákl. deska: MSI K8N Neo4 Platinum
- Osciloskop DSO1004A, Agilent Technologies [31]
- Osciloskop U2702A, Agilent Technologies [31]
- DEMO9S08LC60 [32]
- FITkit [33]
- USB HUB TRUST 7 Port USB2 s napájením HUB HU-5870V
- USB Flash Disk, Transcend, JF V30/GB



Obr. 17: Osciloskop, model DSO1004A.



Obr. 18: Osciloskop, model U2702A.



Obr. 19: FITkit.;



Obr. 20: Zapojení.

5.2 Použité operační systémy

- Arch Linux
- Debian Lenny (5.0.2)
- Fedora 13
- Ubuntu 9.10 „Karmic Koala“ [34]
- Windows XP

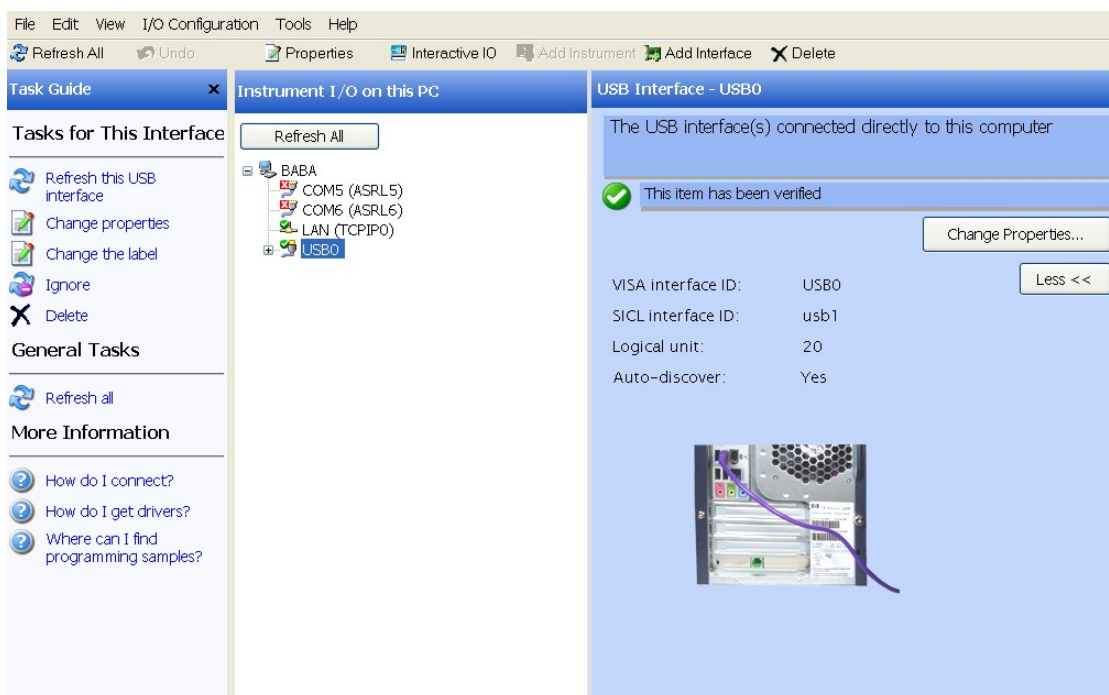
5.3 Software vhodný pro řešení USB over IP problému

- USB/IP project [35]
 - Server: Linux/Windows

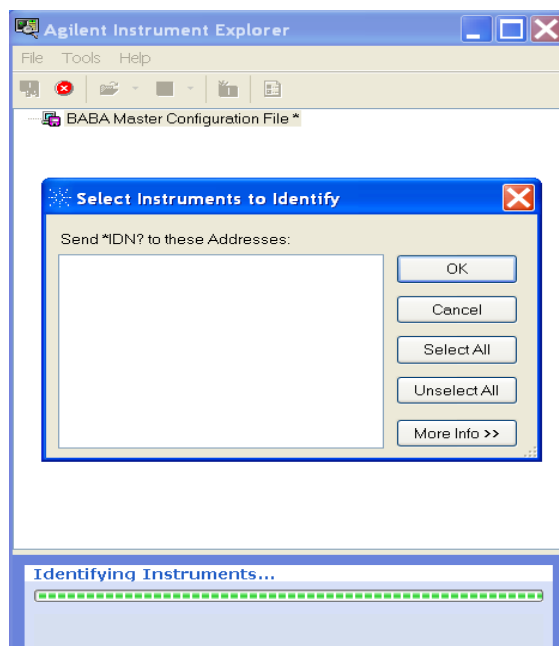
- Client: Linux/Windows
- USB Server [36]
- USB over Network [37]
 - Server: Linux/Windows
 - Client: Windows
- USB to Ethernet Connector [38]
 - eveusb, verze 1.0.2355
 - usb_to_ethernet, verze 4.0.0.513

5.4 Software pužitý k ovládání připojených zařízení

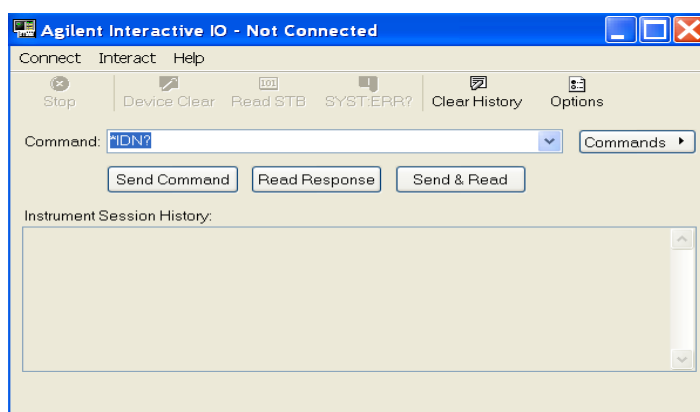
- Agilent Connection Expert, verze 15.5.13009.1
- Agilent Instrument Explorer, verze 2.1.8910.0
- Agilent Interactive IO, verze 2.0.8818.2
- Agilent Measurement Manager, verze 1.8.7.0
- CodeWarrior IDE, verze 5.9.0
- QdevKit, verze 1.1.8



Obr. 21: Agilent Connection Expert.



Obr. 22: Agilent Instrument Explorer.



Obr. 23: Agilent Interactive IO.

5.5 Testování jednotlivých řešení

5.5.1 USB/IP project

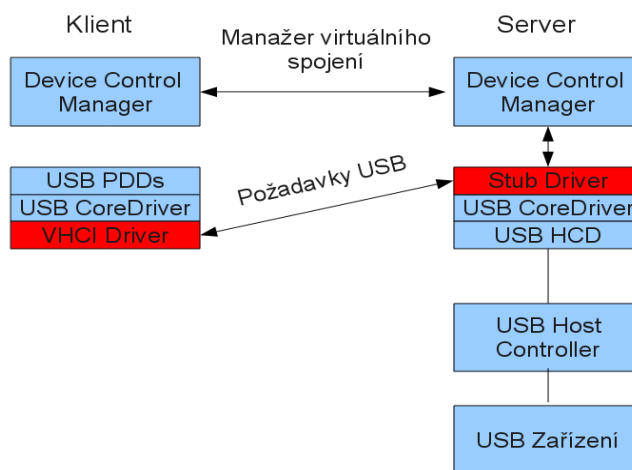
Ke sdílení USB zařízení mezi počítači USB/IP zapouzdřuje „USB I/O zprávy“ do TCP/IP dat a přenáší je po síti mezi počítači. Originální ovladače USB zařízení a aplikace je možné použít bez žádných modifikací. Počítač používá zařízení jako by na něj byla připojena přímo.

Podporovaná zařízení a možná práce s nimi:

- USB paměťová zařízení: fdisk, mkfs, mount/umount, souborové operace, přehrávání DVD filmů a zapisování DVD-R medií.
- USB klávesnice a myši: použití v Linuxové konzoli i v X Windows System.
- USB webkamery a mikrofony: webkamera, zachycení obrazových dat, přehrávání hudby.
- USB tiskárny, USB skenery, USB Ethernet interface-

Popis řešení projektu USB/IP

V klientovi je implementován VHCI ovladač (Virtual Host Controller Interface) jako USB host controller ovladač. VHCI ovladač emuluje reálné USB VHCI pro virtuální připojení/odpojení sdílených USB zařízení.



Obr. 24: USB/IP základní popis. [35]

Server

Instalace USB/IP probíhala na Ubuntu 9.10 „Karmic Koala“, protože má potřebný balíček v repozitáři.

```
# apt-get install usbip
```

Zavedení usbip kernelových modulů:

```
# modprobe usbip
# modprobe usbip_common_mod
```

Spuštění usbip démona:

```
# usbipd -D
```

Výpis zařízení:

```
# usbip_bind_driver -list
```

Nasdílení zařízení:

```
# usbip_bind_driver --usbip <číslo zařízení>
```

Klient

Klient pro Windows se spouští z konzole:



```
C:\Documents and Settings\berus\Dokumenty\usbip-win\usbip-win>usbip.exe
Usage: usbip.exe [options]
  -a, --attach [host] [bus_id]
        Attach a remote USB device.

  -x, --attachall [host]
        Attach all remote USB devices on the specific host.

  -d, --detach [ports]
        Detach an imported USB device.

  -l, --list [hosts]
        List exported USB devices.

  -p, --port
        List virtual USB port status.

  -D, --debug
        Print debugging information.

  -v, --version
        Show version.

  -h, --help
        Print this help.
```

Obr. 25: Ukázka práce s klientem USB/IP.

Seznam přístupných USB zařízení:

```
# usbip -l <ip adresa serveru>
# usbip -l 192.168.1.105
```

Připojení určitého USB zařízení:

```
# usbip -a <ip adresa serveru> <číslo zařízení>
# usbip -a 192.168.1.105 1-1
```

Výsledek testu

USB flash disk šlo připojit a pracovat s ním bez problémů, avšak po připojení osciloskopu jej klient neviděl a nebylo možno s ním proto komunikovat.

5.5.2 USB Server & USB Client for Linux

USB server se bohužel nepodařilo přivést do funkčního stavu – instalace neproběhla korektně.

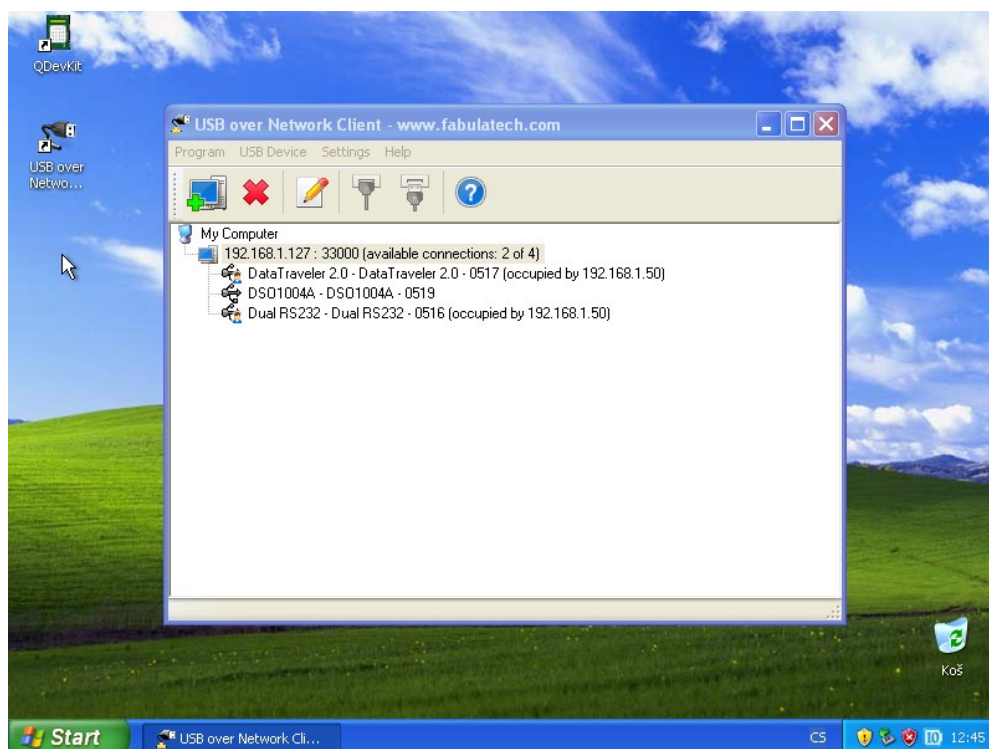
Výsledek testu

Nepodařilo se nainstalovat na daný systém.

5.5.3 USB over Network

USB over Network je výborný produkt společnosti FabulaTech. Produkt se skládá ze dvou částí, serverové a klientské. Instalace je přívětivá a samotná práce s produktem také.

Všechna připojená zařízení fungovala správně a byla rozpoznána, avšak problém nastal při komunikaci pomocí Agilent Measurement Manager, který využívá Agilent T&M Toolkit. Nebylo totiž možné pomocí zmíněného programu komunikovat s připojeným zařízením.



Obr. 26: Ukázka klienta produktu USB over Network.

Výsledek testu

Produkt fungoval správně až na kolizní stav s Agilent T&M Toolkitem, což není fatální záležitost.

5.5.4 USB to Ethernet Connector

USB to Ethernet Connector je produkt společnosti Eltima.

Pro 1 USB zařízení produkt pracoval velmi dobře, proto byla potřeba vyzkoušet toto řešení také pro více zařízení. Společnost poskytla licenci pro 10 USB. Při testování docházelo k problémům s připojením k jednotlivým zařízením, příčina nebyla zjištěna. Problém mohl nastat prodlevami na síti. Produkt se skládá ze dvou částí, serverové a klientské.

5.5.5 Celkové výsledky

Tab. 11: Výsledky testů USB over IP: softwarové řešení.

Výsledky testů	Instalace	USB Flash disk	Jedno zařízení (Osciloskop, FITkit,..)	Více zařízení (Osciloskop, FITkit,..)	Bezkoliznost
USB/IP project	Green	Green	Red	Grey	Grey
USB Server	Red	Grey	Grey	Grey	Grey
USB over Network	Green	Green	Yellow	Yellow	Green
USB to Ethernet Connector	Green	Green	Green	Red	Grey
Test prošel	Green				
Test prošel s výhradami	Yellow				
Test neprošel	Red				
Netestováno	Grey				

5.6 Shrnutí

Testování USB over IP nebylo vůbec jednoduché. Prvním cílem bylo nalézt a otestovat open-source varianty. USB/IP projekt na první pohled vypadal velice slibně a zprovoznitelnost se potvrdila vyzkoušením USB over IP s flash diskem. Avšak testování jednotlivých zařízení, např. osciloskopu nedopadlo dle očekávání, zařízení nebyla rozpoznána. USB Server se nepodařilo ani nainstalovat. Proto bylo nutné porozhlédnout se po komerčních produktech. Jako první byl vyzkoušen USB to Ethernet Connector od společnosti Eltima, který fungoval správně pro jedno zařízení. Bylo potřeba vyzkoušet však pro připojení alespoň pěti zařízení. Po provedení testů se ukázalo toto řešení jako nestabilní, což byla škoda, jelikož cena tohoto řešení je přijatelná. Druhým testovaným komerčním produktem byl USB over Network od firmy FabulaTech. Velmi kvalitní produkt za velmi vysokou cenu. Ve všech testech dopadl dobře, avšak docházelo k problémům při komunikaci se zařízením pomocí Agilent Measurement s využitím Manager Agilent T&M Toolkit, což ale nebyl fatální problém a není jisté, že je problém na straně USB over Network a ne Agilentu. Prováděné testování bylo časově poměrně náročné, jelikož bylo potřeba vyzkoušet zařízení připojit jednotlivě, přistupovat k nim přes testovaný produkt a vyzkoušet si jejich funkčnost. Ta se testovala přes dodávané programy

od výrobce s využitím vzorových příkladů. Dále bylo potřeba otestovat zapojených více zařízení – všechny, co byly k dispozici (uvedené výše v seznamu použitého hardwaru). A nakonec bylo potřeba otestovat přístup z více virtuálních strojů a zjistit, zda nejde k zařízení přistupovat, když je obsazené jiným strojem. Výsledkem testování je, že vývoj řešení USB over IP, které by fungovalo na široký rozsah zařízení, čeká dlouhá cesta, zejména u open-source variant. Jako jediný použitelný produkt je USB over Network, ale jak bylo zmíněno výše, nevýhodou je vysoká pořizovací cena, která se bohužel nevztahuje na neomezený počet zařízení a je vždy limitována počtem připojeným zařízení.

6 USB over IP: Hardwarové řešení

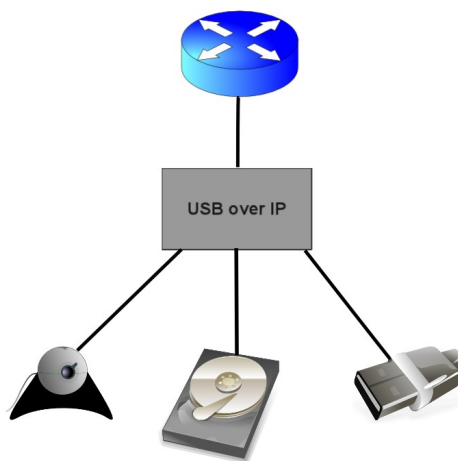
Kromě softwarových řešení existují samozřejmě také hardwarová řešení. Ta fungují ve většině případů jen pro USB flash disky, tiskárny, skenery, apod. Bohužel problém nastává při testování s osciloskopy. Firma Agilent sice nabízí svá zařízení, u kterých by byla jistota, že osciloskopy na nich budou fungovat správně, ale bohužel za poměrně vysokou cenu a také riziko, že nebudou fungovat další zařízení, jenž je potřeba provozovat. Účelem tohoto testu proto bylo vyzkoušet některé levnější varianty.

6.1 Použitý hardware

- USB-OVER-IP zařízení
 - Silex Technology Hi-Speed USB Device Server
 - Belkin Network USB Hub
 - Silex Technology SX-30000 USB Device Server
- Testovací zařízení
 - Externí disk
 - Webkamera
 - USB flash disk

6.2 Postup testování

- Zapojení zařízení dle schématu
- Nainstalování ovládacího softwaru
- Zjištění funkčnosti připojených testovacích zařízení (USB flash disk, externí disk, webkamera).
- Vyhodnocení testů



Obr. 27: Schéma testovacího zapojení.

6.3 Výsledky testování

Výsledky testů dopadly u všech zařízení stejně. USB flash Disk fungoval bez problémů, avšak externí disk a webkamera nebyly rozpoznány, proto nasazení hardwarového USB over IP zařízení bylo zamítnuto.

7 Zabezpečení vzdáleného přístupu

7.1 FreeRADIUS

FreeRADIUS je použit k autentizaci uživatelů jak PPTP VPN tak OpenVPN. [39]

7.1.1 Instalace a konfigurace

Instalace FreeRADIUS serveru a LDAP modulu:

```
# apt-get install freeradius freeradius-ldap
```

Úprava souboru /etc/freeradius/radiusd.conf:

```
...
modules {
...
# Autentizace uzivatelu proti UTB LDAP serveru
ldap ldap_utb {
    server = "nw-central.utb.cz"
    identity = "cn=%{Stripped-User-Name},ou=%{UTB-OU},o=UTB"
    basedn = "cn=%{Stripped-User-Name},ou=%{UTB-OU},o=UTB"
    filter = "(objectclass=*)"
}
# Autentizace uzivatelu proti FIT LDAP serveru
ldap ldap_vut_fit {
    server = "www.fit.vutbr.cz"
    identity = "uid=%{Stripped-User-Name},dc=fit,dc=vutbr,dc=cz"
    basedn = "uid=%{Stripped-User-Name},dc=fit,dc=vutbr,dc=cz"
    filter = "(objectclass=*)"
}
...
}
...

```

Pozn.: Namísto předpokládaných jmen LDAP serverů ldap.utb.cz a ldap.fit.vutbr.cz byla použita jména nw-central.utb.cz a www.fit.vutbr.cz, protože tato jména odpovídají SSL certifikátům daných serverů.

Jména uživatelů mají tvar `uživatel.jednotka@utb.cz` a `uživatel@fit.vutbr.cz`. Část za zavináčem je definována v souboru `/etc/freeradius/proxy.conf`:

```
...
realm utb.cz {
}
realm fit.vutbr.cz {
}
```

Jména UTB uživatelů je potřeba rozdělit na uživatelské jméno a organizační jednotku. Toho lze docílit následující definicí v souboru `/etc/freeradius/hints`.

```
...
DEFAULT User-Name =~ "^(.+)\.\.([\^.]*)@utb\.\.cz$"
        User-Name := "%{1}@utb.cz",
        UTB-OU := "%{2}"
```

Pro uchování organizační jednotky UTB je použit nestandardní RADIUS atribut `UTB-OU`, který musí být definován v souboru `/etc/freeradius/dictionary`:

```
...
ATTRIBUTE          UTB-OU          3000      string
```

Vlastní nastavení serveru se nachází v souboru `/etc/freeradius/sites-enabled/default`:

```
...
authorize {
    preprocess
    suffix
    switch "%{Realm}" {
        case "utb.cz" {
            ldap_utb
        }
        case "fit.vutbr.cz" {
            ldap_vut_fit
        }
        case {
            reject
        }
    }
    update control {
        Auth-Type := LDAP
    }
    expiration
    logintime
    pap
}

authenticate {
    Auth-Type PAP {
        pap
    }
    Auth-Type LDAP {
        switch "%{Realm}" {
            case "utb.cz" {
                ldap_utb
            }
            case "fit.vutbr.cz" {
                ldap_vut_fit
            }
        }
    }
}
...
```

7.2 PPTP VPN

Instalace:

```
# apt-get install pptpd radiusclient-ng
```

7.2.1 PPTP – server

Úprava souboru `/etc/pptpd.conf`, nastavení rozsahu IP adres pro virtuální síť:

```
...
localip 10.11.12.1
remoteip 10.11.12.2-254
...
```

V souboru `/etc/ppp/pptpd-options` nastavit:

```
...
# Encryption
# Debian: on systems with a kernel built with the package
# kernel-patch-mppe >= 2.4.2 and using ppp >= 2.4.2, ...
# {{{
require-pap
#refuse-pap
#refuse-chap
#refuse-mschap
# Require the peer to authenticate itself using MS-CHAPv2 [Microsoft
# Challenge Handshake Authentication Protocol, Version 2] authentication.
#require-mschap-v2
# Require MPPE 128-bit encryption
# (note that MPPE requires the use of MSCHAP-V2 during authentication)
#require-mppe-128
# }}}
...
plugin radius.so
plugin radattr.so
...
```

RadiusKlient

RADIUS plugin PPTP serveru očekává konfiguraci v adresáři `/etc/radiusclient` a také existenci souboru `/etc/freeradius/port-id-map`:

```
# ln -s /etc/radiusclient-ng /etc/radiusclient
# touch /etc/radiusclient-ng/port-id-map
```

Úprava souboru `/etc/radiusclient-ng/radiusclient.conf`

```
...  
auth_order      radius  
...  
authserver      localhost  
acctserver      localhost  
...
```

Úprava souboru `/etc/radiusclient-ng/servers`:

```
localhost      testing123
```

Vytvoření souboru `/etc/radiusclient-ng/dictionary.microsoft` podle [40].

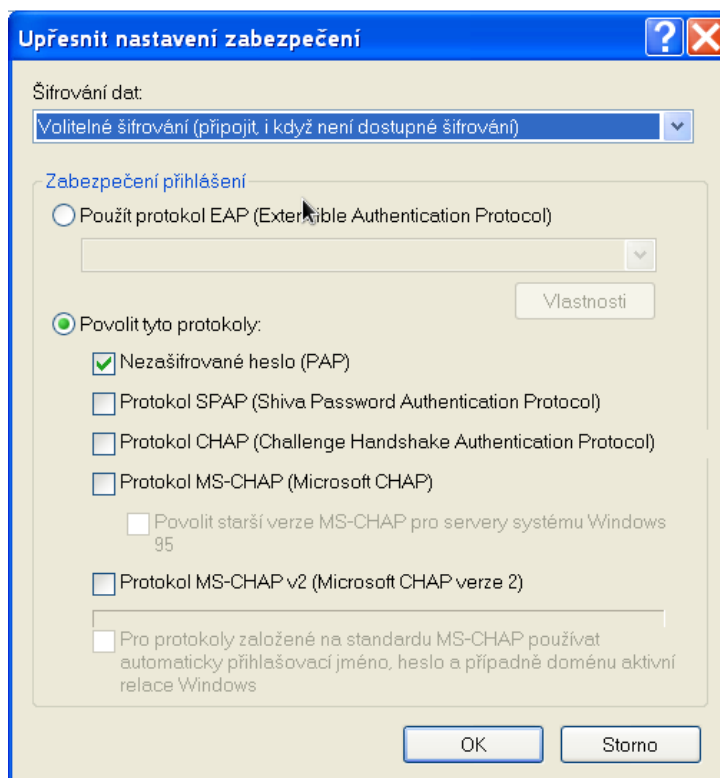
Úprava souboru `/etc/radiusclient-ng/dictionary`

```
...  
INCLUDE        /etc/radiusclient-ng/dictionary.merit  
INCLUDE        /etc/radiusclient-ng/dictionary.microsoft
```

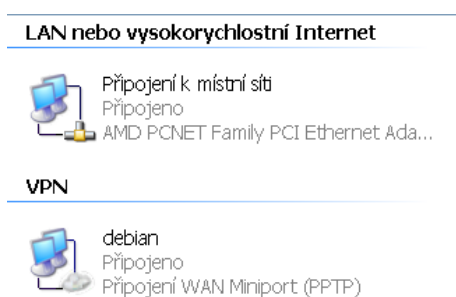
7.2.2 PPTP - klient

MS Windows XP

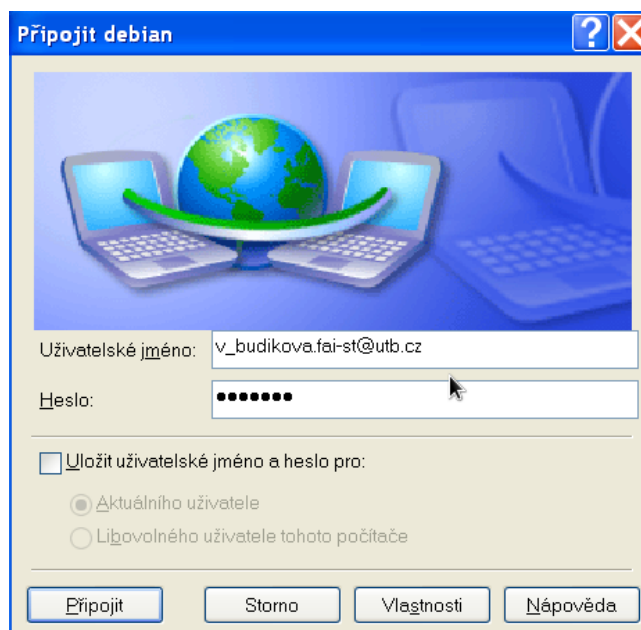
V MS Window XP je klient vestavěný, proto není potřeba žádná instalace, jen jeho nastavení. V sekci síťová připojení se zvolí Vytvořit nové připojení, jako typ připojení se vybere Připojit k firemní síti, připojení k virtuální privátní síti. Dále se zadá název připojení a IP adresa serveru, na který se připojuje. V okně Upřesnit nastavení zabezpečení je potřeba změnit následující:



Obr. 28: Nastavení zabezpečení.



Obr. 30: MS Windows XP - VPN připojení.



Obr. 29: MS Windows XP, vyplnění přihlašovacích údajů.

Linux

V Linuxu je potřeba nainstalovat PPTP klienta. Pro jednodušší ovládání je možnost využití rozšíření NetworkManageru. Nutné je mít nainstalovaný plugin pro podporu PPTP v NetworkManageru.



Obr. 31: Nastavení PPTP VPN, NetworkManager.

7.2.3 Výsledky

Zabezpečení pomocí PPTP VPN je plně funkční, ale vzhledem k použití PAP, ne úplně bezpečná varianta, jelikož není použito šifrování. Další variantou je nasazení OpenVPN, které je popsáno níže.

7.3 OpenVPN

7.3.1 Instalace:

```
# apt-get install openvpn libpam-radius-auth
```

7.3.2 Vytvoření SSL/TLS certifikátů:

```
# cd /usr/share/doc/openvpn/examples/easy-rsa/2.0/
```

Úprava souboru vars:

```
export KEY_COUNTRY="CZ"
export KEY_PROVINCE=""
export KEY_CITY="Brno"
export KEY_ORG="VLAM"
export KEY_EMAIL="budikova@gmail.com"
```

Vytvoření certifikátu certifikační autority a OpenVPN serveru:

```
# . vars
# ./clean-all
# ./build-dh
# ./build-key-server server
# cp keys/ca.crt keys/ca.key keys/server.crt keys/server.key
keys/dh1024.pem /etc/openvpn
# ./clean-all
```

7.3.3 Konfigurace PAM modulu pro přihlašování

Nastavení RADIUS serveru v /etc/pam_radius_auth.conf:

```
localhost testing123 1
```

Nastavení PAM služby v /etc/pam.d/openvpn:

```
auth required pam_radius_auth.so
account required pam_permit.so
```

7.3.4 Konfigurace OpenVPN serveru

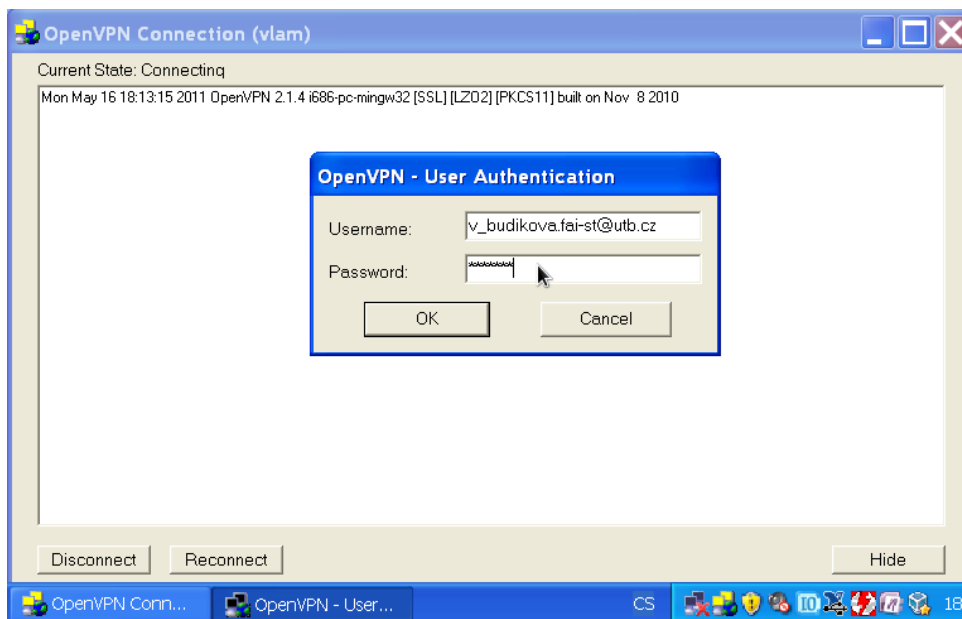
Vytvoření konfiguračního souboru /etc/openvpn/vlam.conf [41]:

```
port 1194
proto udp
dev tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
key /etc/openvpn/server.key
dh /etc/openvpn/dh1024.pem
server 10.8.0.0 255.255.255.0
keepalive 10 120
comp-lzo
user nobody
group nogroup
persist-key
persist-tun
verb 3
client-cert-not-required
username-as-common-name
plugin /usr/lib/openvpn/openvpn-auth-pam.so openvpn
```

7.3.5 Klient – MS Windows XP

Do MS Windows XP je potřeba nainstalovat OpenVPN klienta. Do složky config se zkopíruje certifikát certifikační autority a konfigurační soubor klienta vlam.ovpn:

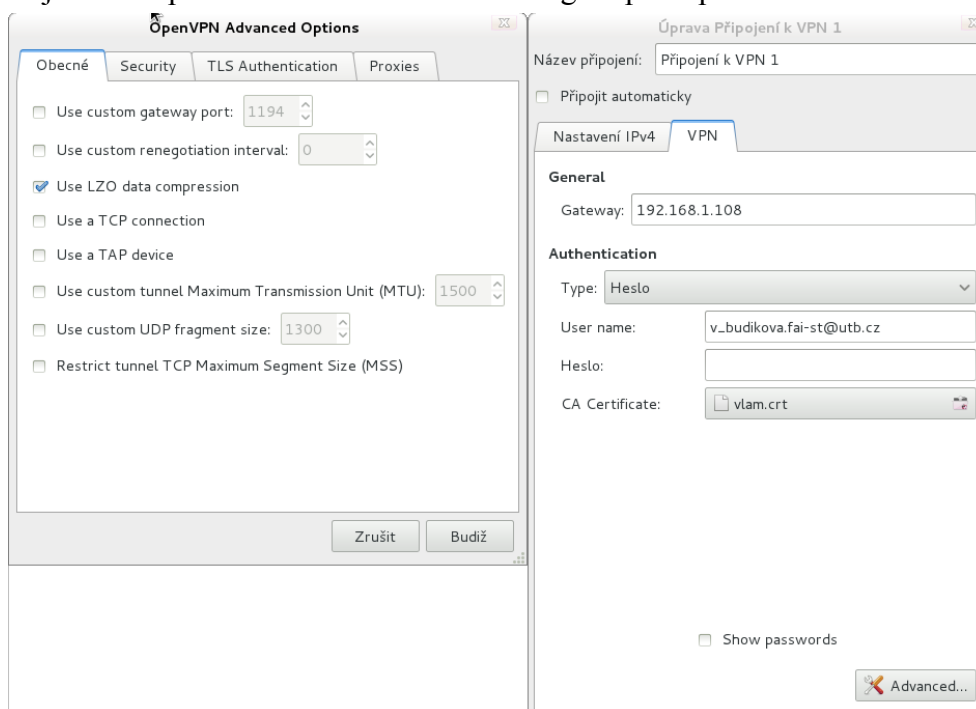
```
remote 192.168.1.108
proto udp
dev tun
ca vlam.crt
client
remote-cert-tls server
comp-lzo
verb 3
auth-user-pass
auth-nocache
```



Obr. 32: OpenVPN, MS Windows XP.

7.3.6 Klient - Linux

V Linuxu je možné použít rozšíření NetworkManageru pro OpenVPN.



Obr. 33: NetworkManager, OpenVPN.

Pro správnou funkčnost je potřeba mít certifikát certifikační autority.

ZÁVĚR

Cílem práce bylo popsat základní strukturu projektu VLAM a navrhnout řešení pro problémy virtualizace, USB přes IP a bezpečnosti vzdáleného přístupu. Dále otestovat RAID 1 a realizovat několik vhodných příkladů pro virtuální laboratoř.

Teoretická část se nejprve zabývá analýzou problému virtuální laboratoře, dále je popsána základní struktura VLAM, zejména serverová část, na kterou se následně navazuje v praktické části. Dále je uveden vyvíjený software v rámci projektu, na jehož vytváření jsem se nepodílela, a to VlamGateway, Vlaxitool, VLAM IDE a PicoBlaze C Compiler. Také jsou zmíněny příklady pro virtuální laboratoř, na jejichž tvorbě se podílela UTIA a které jsou integrovány do VLAM prostředí.

Praktická část obsahuje návrh a testování virtualizační technologie. Na trhu je několik produktů, avšak finální souboj proběhl mezi KVM a Xenem. Pro vybrání nejlepší technologie bylo nutné provést automatické testy, které monitorovaly výkon jednotlivých řešení. Na rozhodování měla vliv také uživatelská přívětivost a náročnost práce s produktem. Po zhodnocení výsledků bylo zvoleno jako nejvhodnější virtualizační nástroj KVM. Ke správné komunikaci se zařízeními, které měly pouze USB port, bylo potřeba nalézt vhodný produkt umožňující USB přes IP. Na výběr bylo jak ze softwarových, tak i hardwarových řešení, bohužel najít fungující produkt je téměř nemožné a pokud vykazoval dobré výsledky, což byl pouze projekt USB over Network, tak byl naneštěstí velmi drahý. Další kapitola se zabývá testováním RAIDu 1. K dispozici byly dvě varianty, softwarový RAID a FakeRAID. Také byla otestována funkce hotswap, která fungovala pro obě provedení RAIDu bez problémů. Nakonec bylo potřeba vyřešit bezpečnost vzdáleného přístupu. Vhodným řešením bylo využití VPN. Nejprve byla zvolena PPTP VPN, protože se předpokládalo, že většina uživatelů používá MS Windows, tedy má vestavěného PPTP klienta a byly by tedy ušetřeny instalace. Při realizaci řešení bylo však zjištěno, že v dané konfiguraci nelze data šifrovat, tedy použití PPTP VPN ztratilo význam. Proto bylo navrženo použití OpenVPN, jenž využívá TLS a data jsou tedy šifrována.

CONCLUSION

The goal of the thesis was to describe the overall structure of the VLAM project and to design a virtualization, USB over IP and secure remote access solution, to test RAID1 and to create several appropriate examples for the virtual laboratory.

The theoretical section first analyzes the problem of virtual lab, then describes the basic structure of the VLAM project, in particular the server part, which the practical section is focused on. Then the software developed as part of the project is mentioned: VlamGateway, Vlaxitool, VLAM IDE and PicoBlaze C Compiler. There are also mentioned examples of the virtual laboratory, whose creation UTIA was involved in and which are integrated into the VLAM environment.

The practical section consists of design and testing of virtualization technology. There are many products on the market, but the final battle took place between KVM and Xen. It was necessary to perform automated tests to monitor the performance of each solution to select the best one. The decision was also affected by the user-friendliness and complexity of each solution. After evaluating the results, KVM was chosen as the best tool for virtualization. For proper communication with the equipment which had only a USB port, it was necessary to find a suitable product to enable USB over IP. The selection was between both software and hardware solutions and, unfortunately, finding a working solution was almost impossible and the only one which showed good results, USB over Network, was unfortunately very expensive. The next chapter deals with testing of RAID1. There were two options, software RAID and FakeRAID. The testing included testing of hotswap, which worked for both implementations of RAID with no problems. Finally, it was necessary to solve secure remote access. A suitable solution was to use VPN. The first selection was PPTP VPN, because it was assumed that most users use Windows, which has built-in PPTP client. However, it was found that the configuration does not permit the data to be encrypted, so the use of PPTP VPN has lost its relevance. It was therefore proposed to use OpenVPN, which uses TLS for data encryption.

SEZNAM POUŽITÉ LITERATURY

- [1] DULÍK, T. *Virtuální laboratoř aplikace mikroprocesorové techniky. Projekt MŠMT č. 2C06008 - prezentace*, 2010 [cit. 2011-05-16].
- [2] GILIBERT, M.; PICAZO, J.; AUER, M.; PESTER, A.; CUSIDO, J.; ORTEGA, J.A. *80C537 Microcontroller Remote Labfor E-Learning Teaching* [online]. 2006 [cit. 2011-05-10]. Dostupné z WWW: <www.inf.u-szeged.hu/~dombi/ai2/Remote.pdf>.
- [3] FERREIRA, J.M.M., NEDIĆ, Z., MACHOTKA, J., NAFALSKI, A., GÖL, Ö. *International collaborative learning using remote workbenchesfor 8-bit microcontroller courses* [online]. 2010 [cit. 2011-05-10]. Dostupné z WWW: <www.wiete.com.au/conferences/1st_wiete/8-21-Ferreira.pdf>.
- [4] *Školní experimentální systém ISES* [online]. 2009 [cit. 2011-05-12]. Dostupné z WWW: <<http://www.ises.info/old-site/default.html>>.
- [5] DULÍK, T., BLIŽŇÁK, M. *Virtual Laboratory of Microprocessor Technology Application*, 2010.
- [6] SUCHÝ, Miroslav *Úvod do virtualizace pomocí XENu* [online]. 2007 [cit. 2010-05-12]. Dostupné z WWW: <<http://www.root.cz/clanky/uvod-do-virtualizace-pomoci-xenu>>.
- [7] BERAN, Radek *Virtualizace operačních systémů* [online]. 2006 [cit. 2011-05-12]. Dostupné z WWW: <<http://www.beranr.webzdarma.cz/virtualizace.html>>.
- [8] *What is OpenVPN?* [online]. 2011 [cit. 2011-05-12]. Dostupné z WWW: <<http://openvpn.net/index.php/open-source/333-what-is-openvpn.html>>.
- [9] BARTOŇ, Martin. *Test: Patriot Valkyrie NAS pro dva disky* [online]. 2010 [cit. 2011-05-12]. Dostupné z WWW: <<http://www.diit.cz/article/print/35986?urltitle=test-patriot-valkyrie-nas-pro-dva-disky>>.
- [10] DULÍK, T. *Linux SW RAID* [online]. 2010 [cit. 2011-05-16]. Dostupné z WWW: <http://wiki.slfree.net/index.php/Linux_SW_RAID>.
- [11] UNIS, UTIA, FIT. *Výroční zpráva 2010 projektu Virtuální laboratoř aplikace mikroprocesorové techniky MŠMT 2C06008*, 2010 [cit 2011-05-02].
- [12] TRBUŠEK, J. *Vzdálené řízení a monitoring měřících přístrojů*. 2011 [cit. 2011-05-10]. 68 s. Diplomová práce. UTB ve Zlíně.
- [13] KADLEC, J., DANĚK, M., SCHIER, J., KOHOUT, L., KAFKA, L., KLOUB, J,

- STEJSKAL, J., SVOZIL, J. Identifikace limitací dosavadních technologií v kontextu projektu. . 2006 [cit. 2011-05-14], , ,
- [14] *Spartan 3E 1600* [online]. 2011 [cit 2011-05-13]. Dostupné z WWW: <<http://www.dfcdesign.cz/obchod/fpga-cpld-moduly/digilent/fpga-moduly/spartan-3e-1600>>.
- [15] *VLAM* [online]. [cit 2011-05-12]. Dostupné z WWW: <<http://www.vlam.cz>>.
- [16] KŘIVKA, Z. , JIRÁK, O. , VAŠÍČEK, J. . *INTEGRATED DEVELOPMENT ENVIRONMENT FOR VIRTUALLABORATORY*, 2011 [cit. 2011-05-12].
- [17] 17: KŘIVKA, Z. , PBCC:PicoBlaze C Compiler User Manual, 2010 [cit. 2011-05-12]
- [18] *Xen* [online]. [cit. 2010-09-09]. Dostupné z WWW: <<http://www.xen.org>>.
- [19] *KVM* [online]. [cit. 2010-09-12]. Dostupné z WWW: <http://www.linux-kvm.org/page/Main_Page>.
- [20] *vmware* [online]. [cit. 2010-08-20]. Dostupné z WWW: <<http://www.vmware.com/cz>>.
- [21] *debian* [online]. [cit. 2010-03-03]. Dostupné z WWW: <<http://www.debian.org>>.
- [22] *fedora* [online]. 2010. Dostupné z WWW: <<http://fedoraproject.org>>.
- [23] WENDELL, John. *Vinagre* [online]. 2011 [cit. 2011-01-26]. Dostupné z WWW: <<http://projects.gnome.org/vinagre>>.
- [24] *Project SIKULI* [online]. 2010 [cit. 2010-09-25]. Dostupné z WWW: <<http://sikuli.org/>>.
- [25] *Python Programming Language* [online]. 2010. Dostupné z WWW: <<http://www.python.org>>.
- [26] *Minimalist GNU for Windows* [online]. [cit. 2010]. Dostupné z WWW: <<http://www.mingw.org>>.
- [27] *Virtualization With KVM On A Debian Lenny Server* [online]. 2010. Dostupné z WWW: <<http://www.howtoforge.com/virtualization-with-kvm-on-a-debian-lenny-server>>.
- [28] *Virtual Machine Manager* [online]. [cit.2010-05-10]. Dostupné z WWW: <<http://virt-manager.et.redhat.com>>.
- [29] MARTIN, Ben *Using Bonnie++ for filesystem performance benchmarking* [online].

- 2008 [cit. 2011-04-28]]. Dostupné z WWW:
<<http://www.linux.com/archive/feature/139742>>.
- [30] *SW RAID - MDADM* [online]. [cit. 2010-05-15]. Dostupné z WWW:
<<http://www.uzitecne.cz/SW-Raid--mdadm/?idfi=2&IDclanku=9>>.
- [31] *Agilent Technologies* [online]. 2010. Dostupné z WWW:
<<http://www.home.agilent.com/agilent/home.jsp?cc=CZ&lc=eng>>.
- [32] *freescale semiconductor* [online]. 2010. Dostupné z WWW:
<<http://www.freescale.com>>.
- [33] *FITkit* [online]. 2010. Dostupné z WWW: <<http://merlin.fit.vutbr.cz/FITkit>>.
- [34] *Ubuntu 9.10 (Karmic Koala)* [online]. 2010. Dostupné z WWW:
<<http://releases.ubuntu.com/karmic>>.
- [35] *USB/IP PROJECT* [online]. 2010 [cit. 2010-09-09]. Dostupné z WWW:
<<http://usbip.sourceforge.net>>.
- [36] *Software Solutions for Sharing and Accessing RemoteUSB Devices over Ethernet* [online]. 2010 [cit. 2010-09-09]. Dostupné z WWW: <<http://www.usb-server.com/products.html>>.
- [37] *USB over the World* [online]. 2010 [cit. 2010-09-09]. Dostupné z WWW:
<<http://www.usb-over-network.com>>.
- [38] *USB to Ethernet Connector* [online]. 2010 [cit. 2010-09-09]. Dostupné z WWW:
<<http://www.eltima.com/products/usb-over-ethernet>>.
- [39] *The FreeRADIUS Project* [online]. [cit. 2011-05-12]. Dostupné z WWW:
<<http://freeradius.org>>.
- [40] *FreeRADIUS WikiPopTop* [online]. 2010 [cit. 2011-05-12]. Dostupné z WWW:
<<http://wiki.freeradius.org/PopTop>>.
- [41] *OpenVPN 2.1* [online]. [cit. 2011-05-12]. Dostupné z WWW:
<<http://openvpn.net/index.php/open-source/documentation/manuals/69-openvpn-21.html>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

API	Application Programming Interface
ASIC	Application-specific-integrated circuit
BIOS	Basic input-output system
BSOD	Blues screen of death
CPU	Central processing unit
DVD	Digital Versatile Disc
FPGA	Field-programmable Gate Array
GCC	GNU Compiler Collection
GNU	GNU is not UNIX
HDL	Hardware Description Language
VLAM	Virtuální laboratoř mikropočítačů se vzdáleným přístupem
HW	Hardware
IBM	Internation Business Machines
IDE	Itegrated Development Environment
IP	Internet Protocol
iSES	Internet School Experimental System
ISO	International Organization for Standardization
KVM	Kernel-based Virtual Machine
LAN	Local Area Network
LCD	Liquid crystal display
LDAP	Lightweight Directory Access Protocol
LED	Light-Emitting Diode
MinGW	Minimalist GNU for Windows
MPGA	Mask-Programmable Gate Array
MS	Microsoft
MSYS	Minimal SYStem
OS	Operating System
PAM	Pluggable authentication modules
PAP	Password Authentication Protocol
PBCC	PicoBlaze C Compiler
PC	Personal Computer
PE	Processr Expert
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
RADIUS	Remote Authentication Dial In User Service
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory

SCPI	Standard Commands for Programmable Instrumentation
SDCC	Small Device C Compiler
SDCCPP	Small Device C Preprocessor
SSH	Secure Shell
SSL	Secure Socket Layer
SW	Software
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
USB	Universal Serial Bus
VHDL	VHSIC hardware description language
VHISC	Very-high-speed integrated circuits
VISA	Virtual Instrument Software Architecture
VLSI	Very-large-scale integration
VNC	Virtual Network Computing
VPN	Virtual Private Network
VPS	Virtual Private Station
VS	Virtual Station

SEZNAM OBRÁZKŮ

Obr. 1: Schéma projektu - 80C537 Microcontroller Remote Lab for	13
Obr. 2: Schéma projektu - 3.2 International collaborative learning using remote workbenches for 8-bit microcontroller courses .[3].....	14
Obr. 3: Architektura projektu VLAM. [5].....	16
Obr. 4: Schéma VPN.....	18
Obr. 5: RAID1. [9].....	20
Obr. 6: Vstup do rezervačního systému VLAM Gateway. [11].....	23
Obr. 7: Okno programu Vlixicon s otevřeným projektem. [12].....	25
Obr. 8: Xilinx Spartan 3E. [14].....	26
Obr. 9: VLAM IDE.[16].....	28
Obr. 10: Architektura PicoBlaze C Compileru (PBCC). [17].....	29
Obr. 11: Prostředí programu Vinagre.....	31
Obr. 12: Ukázka práce s SIKULI.....	32
Obr. 13: Virt-manager, přidání nového připojení.....	37
Obr. 14: Ukázka virtuálních strojů.....	37
Obr. 15: Ukázka práce ve virtuálním stroji pomocí programu Virt-Manager.....	38
Obr. 16: Návrh USB over IP.....	48
Obr. 17: Osciloskop, model DSO1004A.....	50
Obr. 18: Osciloskop, model U2702A.....	50
Obr. 19: FITkit.;.....	51
Obr. 20: Zapojení.....	51
Obr. 21: Agilent Connection Expert.....	52
Obr. 22: Agilent Instrument Explorer.....	53
Obr. 23: Agilent Interactive IO.....	53
Obr. 24: USB/IP základní popis. [35].....	54
Obr. 25: Ukázka práce s klientem USB/IP.....	55
Obr. 26: Ukázka klienta produktu USB over Network.....	56
Obr. 27: Schéma testovacího zapojení.....	59
Obr. 28: Nastavení zabezpečení.....	66
Obr. 29: MS Widnows XP, vyplnění přihlašovacích údajů.....	66
Obr. 30: MS Windows XP - VPN připojení.....	66
Obr. 31: Nastavení PPTP VPN, NetworkManager.....	67
Obr. 32: OpenVPN, MS Windows XP.....	70

Obr. 33: NetworkManager, OpenVPN.....70

SEZNAM TABULEK

Tab. 1: Ekonomický rozbor, 1 lab. pracoviště = 1 plnohodnotné PC. [1].....	12
Tab. 2: Ekonomický rozbor, plně virtualizovaná laboratoř. [1].....	13
Tab. 3: Testování jednoho virtuálního stroje: KVM versus XEN.....	40
Tab. 4: Testování 8 virtuálních strojů - XEN, doba zápisu 100 MB.....	40
Tab. 5: Testování 8 virtuálních strojů - KVM, doba zápisu 100 MB.....	40
Tab. 6: Testování 8 virtuálních strojů - XEN, kB/min.....	41
Tab. 7: Testování 8 virtuálních strojů - KVM, kB/min.....	41
Tab. 8: Výsledky testů virtualizace - subjektivní porovnání z pohledu uživatele.....	41
Tab. 9: Výsledky testů - Bonnie++ (Sequential Output, Input, Random Seeks).....	47
Tab. 10: Výsledky testů - Bonnie++ (Sequential Create, Random Create).....	48
Tab. 11: Výsledky testů USB over IP: softwarové řešení.....	57