

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2009

Petr BERAN

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Úprava firmware na WiFi AP, nahrazení embedded Linuxem

Beran Petr

Bakalářská práce

2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr BERAN**

Studijní program: **B2646 Informační technologie**

Studijní obor: **Informační technologie**

Název tématu: **Úprava firmware na WiFi AP, nahrazení embedded Linuxem**

Z á s a d y p r o v y p r a c o v á n í :

Teoretická část: 1. Vytvořte přehled přístupových bodů (AP) Wi-Fi a jejich funkcí s vhodností použití včetně porovnání ceny. 2. Vyberte levnější AP, kterému lze přehrát firmware a nahradit jej vlastní verzí embedded-Linuxu. Praktická část: 1. Připravte jádro Linuxu pro daný čip AP a pokuste se jej instalovat místo původního firmwaru. Diskutujte problémy a omezení. 2. Pokud se úspěšně povede spustit embedded-Linux na AP, navrhněte vhodný způsob správy a rozšiřování funkcí; pokud se to nezdaří, uveďte příčiny a podmínky úspěšnosti.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

* Kolektiv autorů: **LINUX - dokumentační projekt**. Computer Press 1998 (3. aktualizované vydání), ISBN 80-7226-761-2. * Stones, Richard, Matthew, Neil: **Linux - Začínáme programovat**. Praha: Computer Press, 2000. ISBN 80-7226-307-2. * Stones, Richard, Matthew, Neil: **Linux - Programujeme profesionálně**. Praha: Computer Press, 2001. ISBN 80-7226-532-6. * Reimer, Klaus: **ARM cross-compiling howto** [online]. 2005-08-28 [cit. 2009-01-08]. URL: <http://www.ailis.de/ák/archives/19-ARM-cross-compiling-howto.html> * Shah, S.: **Administrace systému Linux - podrobný průvodce začínajícího administrátora**. Praha: Grada, 2002. ISBN 80-7169-586-6.

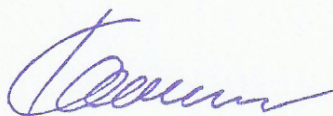
Vedoucí bakalářské práce:

Mgr. Tomáš Hudec

Katedra informačních technologií

Datum zadání bakalářské práce: **15. ledna 2009**

Termín odevzdání bakalářské práce: **15. května 2009**

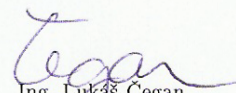


doc. Ing. Simeon Karamazov, Dr.

děkan



L.S.



Ing. Lukáš Čegan
vedoucí katedry

V Pardubicích dne 31. března 2009

Prohlašuji:

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 12. 5. 2009

Petr Beran

Souhrn

Práce se zabývá problematikou nahrazení firmware u embedded zařízení jeho linuxovou alternativou. Obsahuje základní přehled vhodných embedded zařízení, porovnání jejich vlastností z hlediska využití v síťovém provozu. Dále se zabývá výběrem konkrétního zařízení a popisuje následnou realizaci nahrazení linuxovým systémem. V rámci realizace je dokumentován postup tvorby softwarových balíčků vhodného formátu.

Klíčová slova

embedded linux; hardware; kompilace; openwrt

Title

Intranet, the communication and informative portal of company and its significance in business management.

Abstract

The bachelor paper deals with questions of replacing firmware in the embedded device by its Linux alternative. It gives basic overview of suitable embedded device and it compares them from the point of view of usage in the networking. Furthermore, it deals with selection of specific equipment and it describes its subsequent realization of replacing by Linux system. The process of the production of the suitable software is documented within the realization.

Keywords

embedded linux; hardware; compilation; openwrt

Obsah

Souhrn.....	6
Klíčová slova.....	6
Title.....	6
Abstract.....	6
Keywords.....	6
Úvod.....	8
1 Embedded systém.....	9
1.1 Charakteristika.....	9
1.2 Uživatelská rozhraní.....	9
2 Přehled kandidátů.....	10
2.1 Mikrotik Routerboard RB433AH.....	10
2.2 PC Engines Alix3c3.....	12
2.3 Asus WL-520GU.....	14
3 Výběr vhodného kandidáta.....	16
4 Možnosti při nasazování Linuxu na embedded zařízení.....	18
5 Volba postupu pro nahrazení.....	20
6 Charakteristika systému OpenWrt.....	21
6.1 Souborový systém SquashFS.....	22
6.2 Souborový systém JFFS2.....	22
7 Postup realizace.....	24
7.1 Příprava na kompilaci.....	24
7.2 Kompilace pro netboot.....	27
7.3 Konfigurace a kompilace permanentního systému.....	27
7.4 Permanentní uložení systému.....	31
7.5 Počáteční konfigurace.....	35
7.6 Tvorba vlastních balíčků.....	37
Závěr.....	40
Bibliografie.....	41
Seznam literatury.....	41
Seznam tabulek a obrázků.....	43
Seznam použitých zkratk.....	45

Úvod

Prvním moderním vyvíjeným embedded systémem byl naváděcí počítač vesmírného projektu Apollo. Současně prvním, hromadně vyvíjeným embedded zařízením, byl vojenský naváděcí systém D-17 firmy Autonetic pro rakety Minuteman, uvedený v roce 1961. Byl vytvořen z tranzistorů a jako hlavní paměť používal hard disk.

Od té doby našla embedded zařízení uplatnění v široké škále lidských oborů činnosti, ve zdravotnictví, mobilních technologiích, v automobilovém a leteckém průmyslu a podobně. Spolu s vývojem mikroelektroniky se zvyšoval výpočetní výkon a snižovala cena těchto zařízení. Globálně narůstající penetrace připojení k internetu přála také rozvoji embedded zařízení specializovaných pro tuto oblast, nasazení v podobě různých aktivních síťových prvků. Právě takovým zařízením se bude zabývat následující práce. Pojednává o nahrazení dodávaného firmware za jeho linuxovou alternativu.

Práce je koncepčně členěna do několika kapitol. Dojde k vymezení pojmu embedded systém, v další části bude představeno několik kandidátů snadno dostupných na lokálním trhu, z nichž bude vybrán nejvhodnější kandidát pro nahrazení firmware za embedded Linux. Dále bude provedena vlastní realizace permanentního nahrazení za linuxový systém. Na závěr bude dokumentován postup tvorby vlastních softwarových balíků.

1 Embedded systém

1.1 Charakteristika

Embedded systém (vestavěný systém, zabudovaný systém) je jednoúčelový systém, ve kterém je řídicí počítač zcela zabudován do zařízení, které ovládá. Na rozdíl od univerzálních počítačů, jako jsou osobní počítače, jsou embedded počítače většinou jednoúčelové, určené pro předem definované činnosti. Vzhledem k tomu, že systém je určen pro konkrétní účel, mohou tvůrci systému při návrhu optimalizovat pro konkrétní použití a tak snížit cenu výrobku. Embedded systémy jsou zpravidla vyráběny ve velkých sériích, čímž bývá jejich výrobní cena stlačována dolů.

Zařízeními, které spadají do kategorie embedded jsou například kalkulačky, mobilní telefony, herní konzole, ale také mikrovlnné trouby, televizory, různé zdravotnické přístroje, ruční počítače (PDA), navigační systémy, podpůrné systémy v letectví, řídicí jednotky v automobilech, nebo třeba routery.

1.2 Uživatelská rozhraní

Uživatelská rozhraní u embedded systémů mohou nabývat mnoha podob. Od nulového uživatelského rozhraní a po plné rozhraní podobné systému stolního PC. Například v systémech PDA, nebo mobilních telefonů kategorie smartphone. Kompromisem mezi žádným a plným uživatelským rozhraním bývají zařízení, která jsou schopna zobrazovat omezenou množinu znaků a ovládají se pouze několika tlačítky, například kalkulačka nebo rotoped. Hojně používanou možností je síťový přístup přes webové rozhraní, kdy není nutné, aby embedded zařízení obsahovalo zobrazovací jednotku a klávesnici. Další alternativou nevyžadující periferie přímo ovládatelné člověkem je přístup pomocí sériového portu, nebo síťový přístup přes SSH. Některá moderní embedded zařízení nesou standardní konektory vstupu a výstupu, VGA, PS/2, USB. [2]

2 Přehled kandidátů

2.1 Mikrotik Routerboard RB433AH

Jedná se o zástupce ucelené platformy Mikrotik. Tento výrobce dodává na trh celou škálu zařízení specializovaných pro nasazení v síťovém provozu. Může sloužit například jako bezdrátový AP, firewall, nebo směrovač. Jednotlivé modely se od sebe liší výkonem mikroprocesoru, počtem ethernetových portů, množstvím miniPCI slotů a ostatních možnostech hardwarového rozšíření, verzí dodávaného software a v neposlední řadě cenou.

Řada modelů RB 4xx patří do nové, druhé generace a zastupuje tak dosluhující routerboardy řad RB 1xx a RB 3xx. Existuje také verze RB433, která má pomaleji taktovaný procesor a postrádá slot na microSD paměťovou kartu. Inovace spočívají především v rychlejším mikroprocesoru a větším množství operační i NAND paměti. Jádrem systému je mikroprocesor AR7161, který integruje více logických celků:

- jádro cpu,
- dvě 1000 Mbit/s ethernet rozhraní,
- řadič paměti DDR a řadič sériové FLASH paměti,
- 32 bit, 33/66 MHz PCI 2.3 rozhraní,
- rychlé UART rozhraní,
- dvě USB 2.0 rozhraní,
- I²S rozhraní pro přímou podporu externího audio kodeku.

Požizovací cena modelu RB433AH je přibližně 3200 Kč. Včetně nezbytného napájecího zdroje, miniPCI WiFi karty a krabičky je cena do 5000,- Kč s DPH. Proto splňuje kritérium příznivé ceny, lze ho tedy uvažovat při dalším výběru.

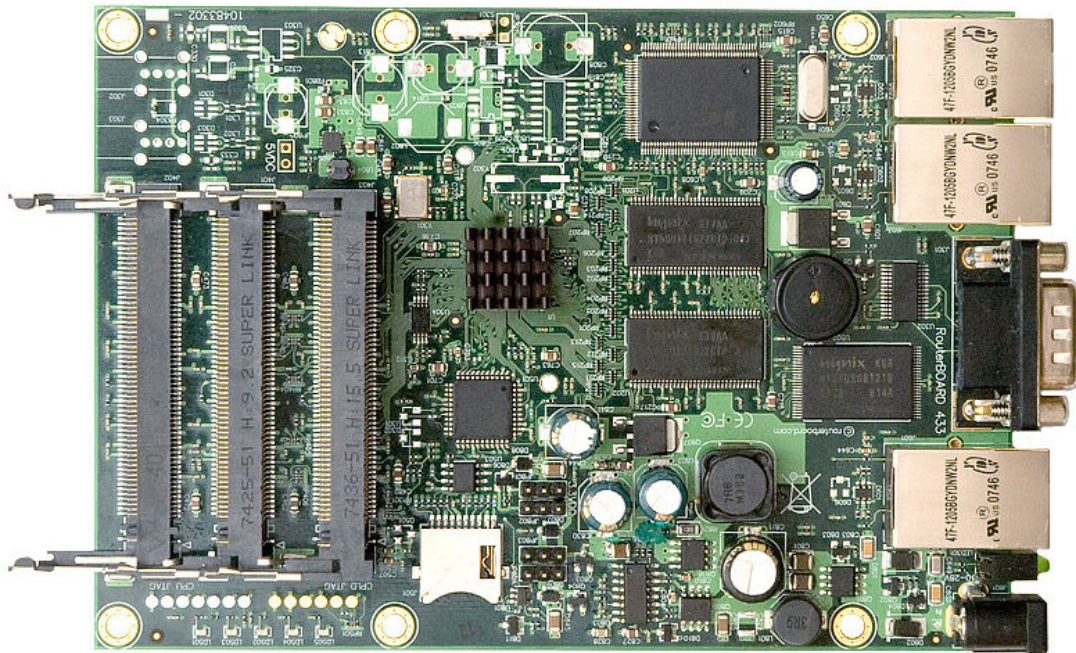
Hardwarová specifikace modelu RB433AH je následující:

- CPU Atheros AR7161 680MHz, architektura MIPS 24k,
- 128 MB DDR SDRAM onboard,
- 64 MB NAND onboard,
- slot pro jednu microSD paměťovou kartu,
- 3x ethernet port 10/100Mbit/s s Auto-MDI/X,
- 3x miniPCI slot,
- rozhraní RS232 – asynchronní sériový port,
- napájecí napětí 10 až 28 voltů, možnost napájet přes ethernet – POE,
- spotřeba bez rozšiřujících karet 3W, maximální spotřeba 25W,
- rozměry 10.5 cm x 15 cm, hmotnost 137 gramů. [3]

Jak bylo řečeno výše, existuje několik verzí licence operačního systému, který je dodáván společně s routerboardem. RB433AH (Ilu. č. 1) je dodáván s licencí RouterOs Level 5. Operační systém RouterOs je closed source aplikace bez zveřejněných zdrojových kódů. Základní rozdíly jednotlivých verzí licence obsahuje následující tabulka.

Vlastnost	Licence	L1 (Free)	L2 (Demo)	L3 (WISP CPE)	L4 (WISP)	L5 (WISP)	L6 (Controller)
Aktualizace		Ne	Ne	ROS v4.x	ROS v4.x	ROS v5.x	ROS v5.x
Podpora		Ne	Ne	Ne	15 dní	30 dní	30 dní
WiFi AP		24h	Ne	Ne	Ano	Ano	Ano
WiFi klient		24h	Ne	Ano	Ano	Ano	Ano
Směr. protokoly		24h	Ne	Ano	Ano	Ano	Ano

Tabulka 1: Přehled verzí licence RouterOs. [4]



Ilustrace 1: Podoba routerboardu Mikrotik RB433AH.

2.2 PC Engines Alix3c3

Alix3c3 je embedded zařízení s univerzálnějším způsobem použití, než předchozí kandidát společnosti Mikrotik. Díky tomu, že je postaven na platformě AMD Geode, je součástí mikroprocesoru také grafické jádro. Deska proto obsahuje standardní VGA konektor pro analogové připojení monitoru. Čip AMD Geode LX800 je zajímavý tím, že integruje v jednom pouzdře více funkčních bloků. Takové čipy se nazývají System-on-chip.

Obsahuje:

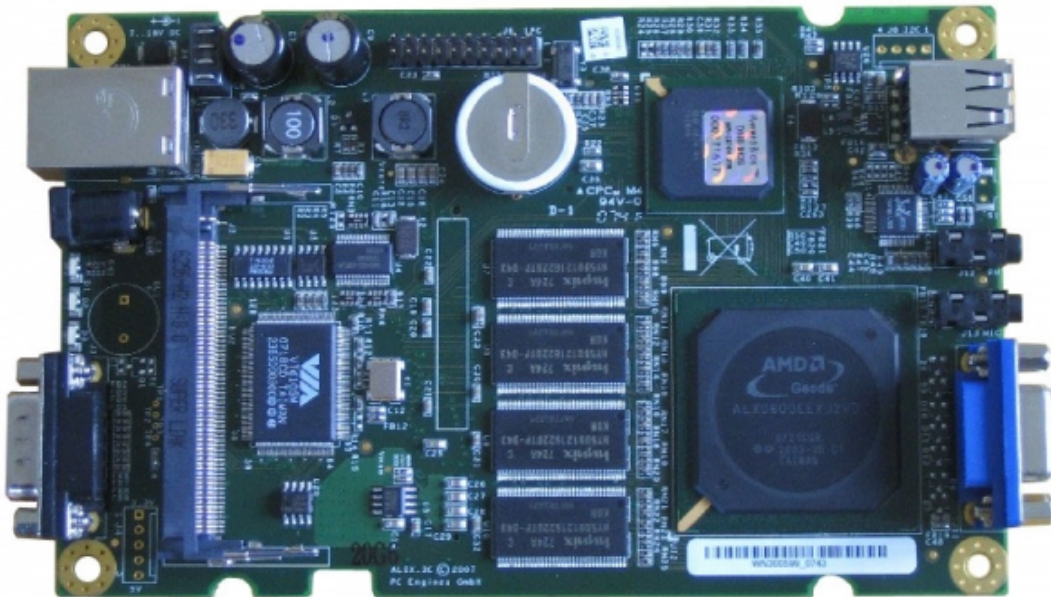
- jádro CPU,
- paměťový řadič,
- grafický procesor,
- logiku pro řízení grafického výstupu,
- hardwarovou podporu 128-Bit AES (Advanced Encryption Standard),
- hardwarový generátor náhodných čísel. [5]

Amd Geode LX 800 se vyznačuje spotřebou 1,8 W, je vyroben 0,13 μm procesem, podporou MMX instrukcí a 3DNow!, podporuje ACPI standard, je dodáván v pouzdře PBGA a má 481 pinů.

Celkově je zařízení koncipováno tak, že může sloužit jako základ terminálové stanice, nebo jako samostatný počítač. Použitelné je také například pro veřejné informační a internetové kiosky, speciální průmyslová zařízení, nebo jako bezdrátový směrovač, či firewall. Stejně jako Mikrotik, dodává společnost PC Engines na trh celou škálu zařízení. Alix3c3 (Il. č. 2) je dodáván bez operačního systému.

Hardwarová specifikace modelu Alix3c3:

- CPU AMD Geode LX800 500MHz, architektura x86,
- 256 MB DDR DRAM onboard,
- jeden slot pro CompactFlash kartu,
- 1x ethernet port 10/100Mbit/s,
- 2x miniPCI slot,
- LPC sběrnice,
- rozhraní RS232,
- 2x USB port,
- 1xVGA výstup,
- dvoukanálový 3.5“ jack audio stereo výstup, 3.5“ jack vstup pro mikrofon,
- napájecí napětí 7 až 20 voltů, možnost napájet přes ethernet – POE,
- spotřeba bez rozšiřujících karet 3W, maximální spotřeba 25W,
- rozměry 10 cm x 16 cm. [6]



Ilustrace 2: Podoba embedded zařízení Alix3c3.

Cena Alix3c3 je přibližně 3500,- Kč s DPH. S krabičkou, napájecím zdrojem, miniPCI WiFi kartou a paměťovou kartou, cena mírně překročí 5000,- Kč s DPH. Pro svoji příznivou cenu, solidní hardwarovou výbavu a univerzální použití je zařazen tento model k dalšímu výběru.

2.3 Asus WL-520GU

Jedná se hardware, který může pracovat jako router, most, nebo bod pro bezdrátové síť standardu IEEE 802.11g/b. Za zmínku stojí možnost používat zařízení jako printserver. Je primárně určen pro domácnost a nenáročné kancelářské použití. Jádrem zařízení je obvod BCM 5354. Obvod je typu System-on-chip a integruje následující komponenty:

- vlastní MIPS procesor,
- 802.11g/b transceiver – vysílací i přijímací část,
- pětiportový 10/100Mbit switch,
- USB 1.1 controller,
- řadič paměti SDRAM. [7]

Pro zařízení WL-520GU (Ilu. č. 3) jsou k dispozici zdrojové kódy dodávaného software. Zařízení se vyznačuje velmi příznivou cenou, která je asi 1600,- Kč s DPH za vše potřebné.

Hardwarová specifikace:

- CPU Broadcom BCM 5354KFBG 240 MHz, architektura MIPS,
- 16 MB SDRAM,
- 4 MB Flash ROM,
- 5x ethernet port 10/100Mbit/s,
- 1x USB 1.1 port,
- RPSMA konektor pro připojení externí antény. [8]



Ilustrace 3: Podoba embedded zařízení WL-520GU.

3 Výběr vhodného kandidáta

Při výběru embedded zařízení vhodného pro další práci je třeba zvážit několik důležitých faktorů. Nemělo by se jednat o exotické zařízení, které postrádá stabilního distributora a z toho plynoucí výhody, například v podobě dostupného servisu. Kvůli způsobu použití zařízení by mělo být možné ho hardwarově rozšířit. Do úvahy spadá hardwarová platforma zařízení, rozšířenost daného zařízení a také cena.

Výše uvedení kandidáti společně splňují kritérium přijatelné ceny a stabilního distributora. Hardwarová platforma je ve dvou případech rodina MIPS a ve zbývajícím x86. I v tomto případě se dá obecně říci, že všechna uvedená zařízení lze stále uvažovat pro další výběr. Operační systém Linux je pro tyto platformy portován. V případě MIPS zařízení je vhodné předem zjistit, zdali existuje podpora pro konkrétní mikroprocesor. Hardwarová modularita má u toho typu zařízení poměrně velkou váhu. Počet bezdrátových zařízení, nebo jiných způsobů spojení bývá často variabilní podle způsobu nasazení. V otázce hardwarové modularity jeden z kandidátů nevyhovuje. Je jím Asus WL-520GU, u kterého je hardwarová modularita nulová. Zařízení má sice velmi výhodnou cenu, ale z dalšího výběru vypadá.

Ve výběru zůstávají dvě zařízení, jedno architektury MIPS a druhé x86. Nyní je třeba opět zvážit hardwarovou modularitu vzhledem ke způsobu použití. Alix3c3 má VGA a zvukový výstup, na desce se také nachází dva USB porty. To je výhodné při použití desky jako tenkého klienta, avšak ve srovnání s routerboardem Mikrotik RB433AH má Alix3C3 menší počet miniPCI a ethernet portů, které považují pro nasazení v síťových aplikacích za důležité. VGA výstup, zvukový výstup a USB porty považují v tomto případě za postradatelné. Mikrotik RB433AH tedy vyhovuje lépe (Tab. č. 2).

Zařízení	HW modularita	HW platforma	Cena s DPH	Dostupnost v ČR
RB433AH	vyhovuje nejlépe	MIPS	3200,- Kč	ano
Alix3c3	vyhovuje	x86	3500,- Kč	ano
WL-520GU	nevyhovuje	MIPS	1600,- Kč	ano

Tabulka 2: Přehled faktorů zohledněných při výběru embedded zařízení.

Na základě předchozího postupu shledávám jako vhodné zařízení pro další práci routerboard Mikrotik RB433AH.

4 Možnosti při nasazování Linuxu na embedded zařízení

Najít vhodný systém

Z hlediska náročnosti přípravy je nejvýhodnější najít systém, který podporuje dané embedded zařízení přímo. Vzhledem k tomu, že jsou embedded zařízení po hardwarové stránce hodně specifická, může být obtížné takový systém najít. Obzvláště u nových kusů hardware je třeba počítat s časovou prodlevou mezi uvedením zařízení na trh a přímou podporou v takovém systému. Výhodou může být komunitní podpora systému, stejně tak existence aktualizací a balíčkovacího systému, který usnadní aktualizaci a řeší závislosti při instalaci softwarových balíčků.

Přizpůsobit distribuci

Další možností je úprava existujícího linuxového systému tak, aby byl schopen provozu na embedded zařízení. Takový postup může být časově náročný a existuje riziko, že se nepovede uvést takový systém do funkčního stavu. Mezi stěžejní patří zajistit podporu pro konkrétní procesor (v případě nasazení na méně rozšířenou hardwarovou platformu), přizpůsobení systému pro případ, že je úložištěm paměť typu FLASH a mnohdy je nestandardní také postup, jakým se nahraje vlastní systém do embedded zařízení.

Linux from scratch

Možnou cestou k úspěšnému nasazení Linuxu na embedded zařízení může být také Linux from scratch. Linux from scratch je postup, kdy dojde k sestavení funkčního systému od základu ze zdrojových kódů. Pro kompilaci se využívá hostitelský systém. V případě sestavení linuxového systému pro jinou platformu, než je platforma hostitelského systému, se používá tzv. cross kompilace. Nevýhodou takového postupu může být fakt, že pomocí Linux from scratch je získána právě taková funkcionálníta, jaká byla zahrnuta při kompilaci. To například znamená, že aktualizace takového systému musí být prováděny ručně, opětovnou kompilací ak-

tuálních verzí software. Použití Linux from scratch je vhodné a mnohdy jediné řešení u velmi hardwarově omezených platform. Linux from scratch má ještě jeden důsledek v podobě absence komunitní podpory pro takový systém. To může být důležitý aspekt při rozhodnutí o nasazení systému.

5 Volba postupu pro nahrazení

Na první pohled se zdálo jako nejlepší řešení využít postup Linux from scratch. Ten poskytne systém ušitý opravdu na míru, avšak daní je relativně náročná aktualizace, pozdější softwarové rozšíření a absence komunitní podpory. Pokud uvažujeme, že bude moci systém Linux from scratch spravovat také někdo další, pokládám za nezbytné přesně dokumentovat celý postup práce. A to navzdory tomu, že Linux from scratch nabízí kvalitní dokumentaci. Proces sestavování je relativně časově náročný a vyžaduje dobrou znalost konkrétního hardware, což nemusí být v praxi ekonomicky únosné. Tento postup jsem tedy ponechal jako možnost v záloze.

Dále jsem se tedy zaměřil na nalezení vhodné distribuce, která by byla tvořena pro embedded zařízení a nabízela by komunitní podporu. Kriteřiem byla možnost detailní konfigurace jádra včetně možnosti výběru požadovaného software. Právě takový systém se mi podařilo najít, jmenuje se OpenWrt.

6 Charakteristika systému OpenWrt

OpenWrt je distribuce která je primárně určená pro embedded zařízení a je založená na linuxu. Původně byla podpora omezena na zařízení Linksys WRT54G series. Postupem času byla přidána podpora pro další chipsety a výrobce hardware. Například Netgear, D-Link, Asus a mnoho dalších. Jako rozhraní primárně využívá příkazový řádek. Volitelně existuje možnost používat GUI rozhraní založené na webovém přístupu.

Existují dvě hlavní verze OpenWrt. První se nazývá *Whiterussian*. Ta funguje pouze na hardware společnosti Broadcom a existuje pro ní kvalitní dokumentace. Vývoj *Whiterussian* byl ukončen ke konci roku 2007. Pro *witerussian* jsou ke stažení tři druhy hotových image systému, bin – označován jako standardní, micro, pro maximální úsporu místa a pptp pro připojení pomocí pptp. Současná verze se nazývá *Kamikaze*. *Kamikaze* se vyznačuje rozšířenou podporou hardware, využívá jádro řady 2.6 pro všechna podporovaná zařízení kromě hardware od Broadcom kvůli zachování funkčnosti bezdrátové části. *Kamikaze* je v současné době stále ve vývoji, využívá pro uložení konfigurace NVRAM, kdežto *Kamikaze* používá konfigurační soubory. [9]

Distribuce OpenWrt využívá systém správy balíčku Ipkg. Ipkg – Itsy package management system – je balíčkovací systém pro instalace software z repozitářů. Primárně je vytvořen pro embedded zařízení, kdy se při menší hardwarové náročnosti snaží přiblížit vlastnostem balíčkovacího systému distribuce Debian. Pokud je systém sestaven pro embedded zařízení, které data uchovává v paměti typu FLASH, OpenWrt využívá dva souborové systémy. SquashFS a JFFS2.

Systém je distribuován buď jako zkompileovaný image s předem určenou množinou možností, nebo ve formě zdrojových kódů pro možnost kontrolovat výslednou funkčnost. Existuje rovněž framework pro kompilaci aplikací, aniž by bylo třeba kompilovat znovu celý systém.

6.1 Souborový systém SquashFS

Jedná se o souborový systém oficiálně podporující kompresi GZIP, ta má podporu přímo v linuxovém jádře. Volitelně podporuje také kompresi LZMA. Maximální velikost souboru je 2^{64} bytů. SquashFS je read-only souborový systém. Využívá se také například jako souborový systém pro LiveCD. Read-only vlastnost SquashFS umožňuje konfigurovat OpenWrt pouze pomocí změn v NVRAM. Neumožňuje tedy přímo měnit soubory a proto se příliš nehodí pokud chceme využívat komplexnější aplikace.

6.2 Souborový systém JFFS2

OpenWrt *Kamikaze* implicitně pracuje na souborovém systému JFFS2. JFFS2 – Journalling Flash File System version 2 – je souborový systém uzpůsobený pro použití na pamětech typu FLASH a Je součástí linuxového jádra od verze 2.4.10.

Hlavní rysy JFFS2:

- přizpůsoben pro běh na zařízeních typu FLASH,
- podporuje hard linky,
- existuje možnost využití kompresních algoritmů zlib, rubin a rtime,
- zlepšuje výkon oproti předchozí verzi.

JFFS2 je žurnálovací souborový systém, který obsahuje v podstatě pouze žurnál a nikoli samostatné soubory. V principu systém pracuje tak, že se poprvé soubor zapíše do žurnálu celý. Při zapisování změny soubor zůstane nedotčen a změna se zapíše opět do žurnálu. Při čtení se soubor s pomocí žurnálu rekonstruuje. Při mazání souboru se do žurnálu zapíše změna, že soubor byl smazán, ale ve skutečnosti se stále nachází na původním místě. Při tomto přístupu vlastně vznikají místa plná dat, která nelze využít pro další zápis. Tento problém řeší garbage collector, který vytvoří ze žurnálu soubor včetně změn a ten opět zapíše do FLASH. V případě že je soubor označen za smazaný, uvolní garbage collector patřičné místo pro další zápis garbage collector se spouští, když dojde k zaplnění jednotky.

Z tohoto způsobu práce se soubory vyplývají některé důsledky:

- i když smažeme soubor, je třeba zápis, může se tak stát, že když dojde volné místo, nelze ani mazat. To souborový systém ošetřuje tak, že nechává několik sektorů obsazených za účelem mazání, nebo spuštění garbage collectoru,
- díky způsobu práce se systémem souborů dochází k rovnoměrnému opotřebení FLASH paměti.

Díky tomu, že JFFS2 využívá pro uložení dat kompresi, nelze přesně určit, kolik volného místa zbývá na diskovém oddílu. Při mountování oddílu jsou všechny nody skenovány, to může trvat dlouhou dobu a tato doba se prodlužuje společně se vzrůstající velikostí diskového oddílu.

7 Postup realizace

Postup realizace se bude skládat z několika dílčích kroků. Jsou jimi:

- příprava na kompilaci,
- kompilace systému pro zavedení systému po síti,
- konfigurace a kompilace systému pro FLASH disk – tento nahradí původní firmware routerboardu,
- permanentní uložení systému do paměti routerboardu,
- počáteční konfigurace.

7.1 Příprava na kompilaci

Rozhodl jsem se vydat cestou vlastní kompilace, což vyžaduje hostitelský linuxový systém. Největší zkušenosti mám s distribucí Ubuntu, proto ho volím jako hostitelský systém. Bude použita aktuální verze 8.10 „Intrepid Ibex“.

Prvním a nezbytným krokem je získání zdrojových kódů pro kompilaci systému. Nejprve vytvoříme adresář pro stažení souborů, nejlépe v domovském adresáři:

```
mkdir ~/openwrt  
cd ~/openwrt
```

Pro mikroprocesor AR71xx je doporučeno stáhnout poslední, vývojovou verzi zdrojových kódů. OpenWrt pro vývoj používá systém pro správu subverzí SVN. Aplikace SVN není ve výchozím stavu v hostitelské distribuci přítomna, doinstalujeme ji příkazem:

```
sudo apt-get install svn  
svn co https://svn.openwrt.org/openwrt/trunk/
```

Tou byla v době realizace revize 15054, se kterou se mi bohužel nepodařilo OpenWrt zprovoznit. Později se jako funkční ukázala starší revize 13577. Pro její získání jsem použit intuitivní program s grafickým rozhraním RapidSvn.


```
sudo apt-get install rapidsvn
```

OpenWrt používá pro konfiguraci a nastavení parametrů jádra Buildroot (Ilu. č. 4). Buildroot je sada makefile a patchů, díky kterým snadno získáme jednak samotný systém v podobě vhodného image pro naše embedded zařízení, ale také cross-compilation toolchain, který nám umožní později kompilovat vlastní programy pro jinou, než hostitelskou hardwarovou platformu. V našem případě je hostitel x86 kompatibilní a cílový systém je architektury MIPS.

Úspěšné spuštění buildrootu vyžaduje následující balíky:

- ncurses – knihovna poskytující rozhraní pro tvorbu aplikací,
- gawk – univerzální jazyk pro zpracování textových dat,
- flex – fast lexical analyzer generator.

Instalaci provedeme příkazem:

```
sudo apt-get install gawk flex libncursesw5-dev
```

Po instalaci těchto balíků lze spustit buildroot příkazem:

```
make menuconfig
```



Ilustrace 4: Konfigurační prostředí buildroot

7.2 *Kompilace pro netboot*

Pro kompilaci image určeného pro bootování ze sítě není třeba se zabývat konfigurací. Tento image poslouží pouze jako mezikrok k permanentnímu uložení později konfigurovaného systému na routerboard. Využijeme tak výchozí výběr buildrootu. V tomto kroku je důležité vybrat pouze správně podporovanou platformu a formát, do jakého bude výsledný image přeložen. V buildrootu nastavíme následující položky:

Target system → Atheros AR71xx[2.6]

Target Images → ramdisk

Buildroot ukončíme klávesou ESC, uložíme změny a spustíme vlastní kompilaci následujícím příkazem:

```
make
```

Na osobním počítači s procesorem Athlon XP 2600+ a 1GB operační paměti kompilace trvala přibližně 40 minut. Jejím výstupem je soubor `openwrt-ar71xx-vmlinux-initramfs.elf`, který je vhodný pro prvotní naboootování systému po síti. Je umístěn v podsložce `bin`.

7.3 *Konfigurace a kompilace permanentního systému*

Po úspěšné předchozí kompilaci můžeme přejít k detailní konfiguraci a opětovné kompilaci. Bude ponechána výchozí konfigurace buildrootu s přihlédnutím na použití routerboardu v síťovém provozu. Spustíme tedy znovu buildroot a přidáme následující položky menu:

```
make menuconfig
```

Nejdříve změníme v sekci Target Images formát výsledného image systému na `tgz`. Později bude uložen do FLASH paměti routerboardu. [11]

Target Images → `tgz`

Sekce *Image configuration* umožňuje nastavit výchozí parametry protokolu IPv4 pro první lan rozhraní routerboardu. Ponecháme implicitní hodnoty 192.168.1.1/24.

Sekce Base system

Obsahuje základní soubory a nástroje systému. Vybereme následující položky:

- iptables – nástroj pro administraci firewallu, vybereme všechny vnořené možnosti.

V této sekci se také nachází nastavení BusyBoxu.

Podsekce Busybox

BusyBox je open-source aplikace, která poskytuje mnoho standardních unixových nástrojů, blíží se větším a komplexnějším GNU Core Utilities. Díky příznivé velikosti je používán pro speciálně zaměřené linuxové distribuce a embedded zařízení. Jeho konfigurace obsahuje další podsekce.

Coreutils:

- nice – spouští programy s upravenou prioritou,
- usleep – pozastaví program na určenou dobu, hodí se při psaní skriptů.

Login/Password Management Utilities:

- Support for shadow passwords – zvyšuje bezpečnost systému při pokusu o získání hesla uživatele.

Linux Module Utilites:

- modprobe – nahraje modul běžícího do jádra systému.

Networking utilities:

- Enable IPv6 support – poskytne podporu pro protokol IPv6,
- arp – pro získání fyzické adresy na základě známé logické adresy,
- brctl – poskytuje možnost konfigurovat ethernetová rozhraní jako bridge, využitelné pro spojování sítí do větších celků,
- ifconfig – používá se pro konfiguraci síťových rozhraní,
- ip – pro konfiguraci TCP/IP a routování,
- ipcalc – nástroj pro výpočty s ip adresami,

- nameif – přejmenuje systémový název síťového rozhraní,
- netstat – informace o síťových spojeních,
- tftp – klient protokolu FTP,
- tftpd – server protokolu FTP,
- traceroute – slouží k analýze počítačové sítě, vypisuje uzly na cestě paketu k cíli,
- udhcpd – dhcp server původně určený pro embedded zařízení.

Process Utilities:

- watch – slouží k periodickému spouštění programu.

Sekce Network:

- iw – slouží pro konfiguraci bezdrátových síťových rozhraní,
- tc – utilita pro kontrolu síťové propustnosti,
- wpa supplicant – podpora pro wpa supplicant.

Kernel modules

Tato sekce slouží k detailní konfiguraci linuxového jádra.

Netfilter Extensions:

- imq - Je virtuální zařízení, kam se přesměruje veškerý provoz (nebo provoz, který chceme ovlivňovat) a tím pádem dostaneme zařízení se vstupem či výstupem a můžeme na něm cokoli ovlivňovat,
- vybereme všechny možnosti jako moduly.

Filesystems:

- kmod-fs-ext2 – podpora pro ext2 souborový systém,
- kmod-fs-ext3 – podpora pro ext3 souborový systém,
- kmod-fs-minix – podpora pro minix souborový systém,
- kmod-fs-vfat – podpora pro FAT32 souborový systém.

Network Support:

- kmod-gre – generic routing encapsulation – protokol ze skrupiny TCP/IP určený k zapouzdření paketů jednoho protokolu do jiného. Používá se pro VPN, tunelování IPv6 přes IPv4 a k tunelování obecně,
- kmod-ipsec – podpora pro Ipsec – to je bezpečnostní rozšíření IP protokolu,
- kmod-iptunnel4 – podpora IPv4 tunelování,

- kmod-iptunnel6 – podpora IPv6 tunelování,
- kmod-tun/tap – podpora tun/tap virtuálních rozhraní,
- kmod-sit – pro tunelování IPv6 přes IPv4.

Other modules:

- kmod-mmc – podpora pro MMC/SD paměťové karty, routerboard obsahuje slot pro jednu microSD paměťovou kartu, může tedy najít využití.

Wireless Drivers:

- je výhodné vybrat celou sekci, obsahuje podporu pro nejrůznější bezdrátové chipsety.

Jelikož nevíme, k jakému konkrétnímu účelu bude routerboard použit, může se později stát, že bude některá funkcionality chybět. Proto byla ponechána implicitní konfigurace buildrootu a byly přidány položky, které by mohly být potenciálně třeba. Popis implicitních položek buildrootu není uveden záměrně, neboť cílem nebylo popsat každou jednu volbu konfigurace. Záměr byl v poukázání na způsob konfigurace buildrootu s přihlédnutím k nasazení pro síťový provoz. OpenWrt je poměrně dynamický projekt, některé vlastnosti přibývají, jiné vypadnou z implicitní konfigurace a proto by takový výčet neměl téměř žádnou informační hodnotu. Pro ponechání implicitní konfigurace také nahrává fakt, že routerboard RB433AH není nijak extrémně hardwarově omezen.

Po uložení změn v buildrootu spustíme opětovnou kompilaci příkazem:

```
make
```

Druhá kompilace trvá kratší dobu, neboť spousta položek je zkompileována již z první kompilace. Druhá kompilace trvala na stejném stroji přibližně 10 minut. Jejím výstupem je soubor `openwrt-ar71xx-vmlinux.elf` – linuxové jádro o velikosti 2,2 MB a `openwrt-ar71xx-rootfs.tgz` – root filesystem o velikosti 3,9 MB. Soubory se nachází v podadresáři `bin`.

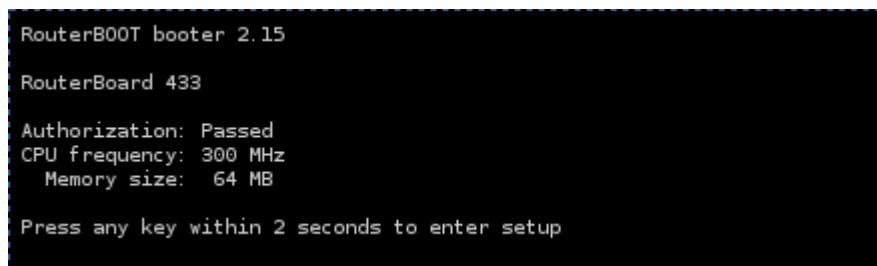
7.4 Permanentní uložení systému

Tato fáze vyžaduje následující kroky. Propojení routerboardu s PC, nastavení bootování po síti, smazání původního systému a permanentní uložení jeho linuxové alternativy. [11]

Počáteční spojení a nastavení realizujeme přes vestavěné rozhraní sériového portu RS232. Využijeme aplikaci cu, v terminálu použijeme tento příkaz:

```
cu -l ttyS0 -s 115200
```

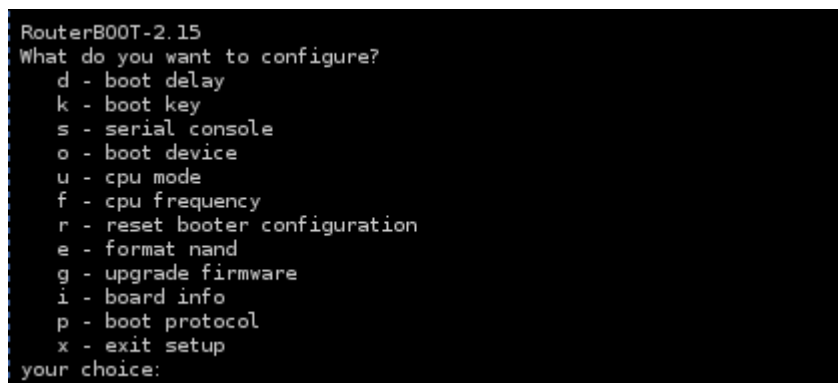
Parametr -l říká, který sériový port na hostitelském pc bude použit a parametr -s určuje rychlost komunikace v bitech za sekundu. Pokud je spojení úspěšné, routerboard začne okamžitě bootovat (Ilu. č. 5):



```
RouterB00T booter 2.15
RouterBoard 433
Authorization: Passed
CPU frequency: 300 MHz
Memory size: 64 MB
Press any key within 2 seconds to enter setup
```

Ilustrace 5: Bootovací obrazovka Mikrotik RB433AH.

Stiskem libovolné klávesy během prvních dvou vteřin startu aktivujeme bootup menu (Ilu. č. 6), kde nastavíme bootování ze sítě a smažeme původní systém ve FLASH paměti routerboardu.



```
RouterB00T-2.15
What do you want to configure?
d - boot delay
k - boot key
s - serial console
o - boot device
u - cpu mode
f - cpu frequency
r - reset booter configuration
e - format nand
g - upgrade firmware
i - board info
p - boot protocol
x - exit setup
your choice:
```

Ilustrace 6: Bootup menu Mikrotik RB433ah.

Boot ze sítě nastavíme v sekci boot device, klávesa „o“, následně klávesou „e“ vybereme ethernet boot. Původní firmware smažeme klávesou „e“ – format nand. Pro uložení a restart stiskneme klávesu „x“. Tak je smazaný routerboard připraven pro boot ze sítě.

Vlastní bootovací proces vyžaduje také správně nastavit hostitelský systém. Nastavení spočívá ve správné konfiguraci DHCP a TFTP serveru. DHCP protokol umožní routerboardu při startu získat předem definované nastavení protokolu TCP/IP, TFTP server zajistí následný přenos při bootování systému. Plně vyhoví DHCP server obsažený v balíku Dnsmasq, pro TFTP pak použijeme balík tftpd-hpa.

```
sudo apt-get install dnsmasq
sudo apt-get install tftpd-hpa
```

Nejprve nakonfigurujeme DHCP server, jeho konfigurace je v souboru /etc/dnsmasq.conf, upravíme je do následující podoby:

```
#nastaveni rozsahu DHCP serveru
    dhcp-range=192.168.6.100,192.168.6.119

#prideleni konkretni ip adresy dane MAC adrese
#rozhrani na routerboardu
    dhcp-host=00:0c:42:40:77:58,192.168.6.101

#lokace souboru pro netboot, adresa rozhrani, pro
ktere bude soubor k dispozici
    dhcp-boot=/tftpboot/openwrt-ar71xx-vmlinux-
initramfs.elf,boothost,192.168.6.1
```

Soubor openwrt-ar71xx-vmlinux-initramfs.elf jsme získali při první kompilaci, poslouží pro boot ze sítě. Je důležité, aby se soubor nacházel na místě, které je v konfiguračním souboru, tzn. /tftpboot.

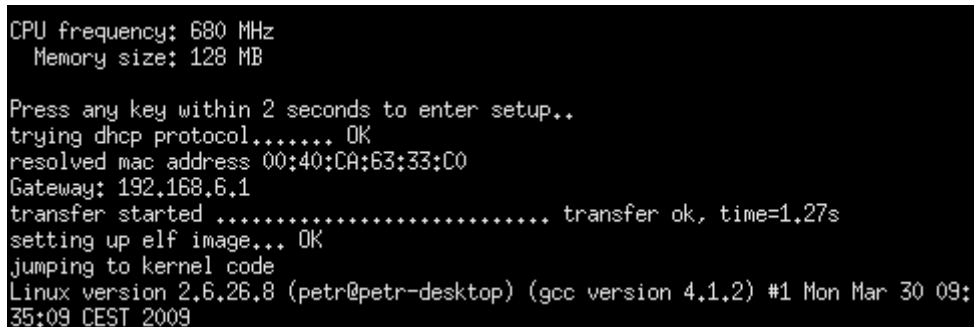
V tomto momentě propojíme routerboard a hostitelský počítač přímo kříženým ethernetovým kabelem, nebo přes síťový switch či hub. Po spuštění obou služeb můžeme zapnout routerboard. Pokud je vše nastaveno správně routerboard okamžitě bootuje ze sítě. Spuštění TFTP a DHCP provedeme takto:

```
sudo /etc/init.d/tftpd-hpa start
sudo /etc/init.d/dnsmasq restart
```

V případě, že routerboard nemůže získat soubor s image systému, řekneme TFTP serveru, aby běžel v režimu standalone:

```
sudo in.tftpd -l
```

System startuje ze sítě (Ilu. č. 7).



```
CPU frequency: 680 MHz
Memory size: 128 MB

Press any key within 2 seconds to enter setup..
trying dhcp protocol..... OK
resolved mac address 00:40:CA:63:33:C0
Gateway: 192.168.6.1
transfer started ..... transfer ok, time=1.27s
setting up elf image... OK
jumping to kernel code
Linux version 2.6.26.8 (petr@petr-desktop) (gcc version 4.1.2) #1 Mon Mar 30 09:35:09 CEST 2009
```

Ilustrace 7: Počátek úspěšného bootu po síti.

OpenWrt implicitně nastavuje TCP/IP protokol prvního ethernet rozhraní na 192.168.1.1. V dalším kroku je třeba mít routerboard a hostitelský systém v jedné síti. Hostitelský systém je v síti 192.168.6.0, routerboard tedy nastavíme do stejné sítě:

```
ifconfig br-lan 192.168.6.2 netmask 255.255.255.0
```

Správnost nastavení ověříme příkazem ping ze systému Openwrt:

```
ping 192.168.6.1
```

Poslední krok k úspěšnému nahrazení distribucí OpenWrt je nahrát permanentně systém do FLASH paměti routerboardu. Systém se přeneseme příkazem `wget2nand`, úspěšný přenos vyžaduje funkční http server na straně hostitelského PC. Plně vyhoví micro-http server obsažený v balíku `micro-httpd`, doinstalujeme jej do hostitelského PC:

```
sudo apt-get install micro-httpd
```

Cesta ke konfiguračnímu souboru je `/etc/inetd.conf`.

```
www      stream  tcp      nowait  root
/usr/sbin/tcpd      /usr/sbin/micro-httpd /tftpboot
#do složky /tftpboot zkopírujeme oba soubory
ziskane pri druhe kompilaci, openwrt-ar71xx-
vmlinux.elf a openwrt-ar71xx-      rootfs.tgz
```

Systém do routerboardu přeneseme následujícím příkazem, kde IP adresa patří hostitelskému PC:

```
wget2nand http://192.168.6.1
```

OpenWrt zpřístupňuje interní FLASH paměť routerboardu přes MTD framework. Paměť je rozdělena do tří oddílů, `mtd0`, `mtd1` a `mtd2`. `Mtd1` obsahuje bootloader, `mtd2` linuxový kernel a `mtd3` root filesystem. Příkaz `wget2nand` provede právě to, že stáhne jádro OpenWrt spolu s root filesystemem z hostitelského PC a nahraje je do správných oddílů. Mtd framework je použit zjednodušeně proto, že Linux umí pracovat pouze se dvěma typy zařízení, `block device` a `character device`. Problém je v tom, že paměť typu FLASH nelze správně zařadit ani do jedné kategorie. FLASH paměť je spíše podobná kategorii `block device`, avšak má některé odlišnosti (Tab. č. 3):

passwd

Trvalé nastavení ethernet rozhraní eth0 zaneseme do konfiguračního souboru /etc/config/network (Ilu. č. 9). K dispozici je editor vi. Struktura konfiguračního souboru je na obrázku níže:

```
root@OpenWrt:/etc/config# cat network

config 'interface' 'loopback'
    option 'ifname' 'lo'
    option 'proto' 'static'
    option 'ipaddr' '127.0.0.1'
    option 'netmask' '255.0.0.0'

config 'interface' 'lan'
    option 'ifname' 'eth0'
    option 'type' 'bridge'
    option 'proto' 'static'
    option 'ipaddr' '192.168.1.20'
    option 'netmask' '255.255.255.0'
    option 'dns' '192.168.1.254'
    option 'gateway' '192.168.1.254'
```

Ilustrace 9: Ukázka struktury konfiguračního souboru pro ethernetové rozhraní.

Z důvodu změny struktury webu OpenWrt opravíme url k repozitářům. Konfigurační soubor se nachází v /etc/opkg.conf:

```
root@OpenWrt:/etc# cat opkg.conf
src/gz snapshots http://downloads.openwrt.org/kamikaze/8.09/ar71xx/packages
dest root /
dest ram /tmp
lists_dir ext /var/opkg-lists
```

Ilustrace 10: Konfigurační soubor pro balíčkovací systém.

V případě přístupu routerboardu do internetu můžeme plně využívat balíčkovací systém ipkg. Například:

- ipkg update – aktualizuje seznam balíků,
- ipkg install <nazev baliku> – instaluje balík,
- ipkg remove <nazev baliku> – odinstaluje balík,
- ipkg upgrade – aktualizace všech balíku na aktuální verze.

7.6 Tvorba vlastních balíčků

Tvorba vlastních balíčků vyžaduje získat SDK pro OpenWrt. SDK je ke stažení na stránkách projektu. Získáme jej následujícím způsobem:

```
cd ~
wget      http://downloads.openwrt.org/whiterussian/
newest/OpenWrt/SDK-Linux-i686-1.tar.bz2
bzipcat OpenWrt-SDK-Linux-i686-1.tar.bz2 | tar -xvf -
cd ~/OpenWrt-SDK-Linux-i686-1
```

Pro příklad budeme vytvářet balík z klasické aplikace psané v jazyku C, helloworld. OpenWrt SDK pracuje tak, že projde všechny složky balíčků a zavolá jejich soubor makefile. Složky balíčků se nachází v podadresáři package. Vytvoříme v něm tedy složku našeho budoucího balíčku:

```
cd package
mkdir hello
```

V adresáři hello vytvoříme soubor pojmenovaný main.c s následujícím zdrojovým kódem:

```
#include <stdio.h>
int main(){ printf("Hello World\n");return 0; }
```

K souboru main.c vytvoříme soubor makefile s následujícím obsahem, aby došlo k úspěšné kompilaci, je nutné správně používat formátovací znaky textu:

```
include $(TOPDIR)/rules.mk

PKG_NAME:=hello
PKG_VERSION:=1.0
PKG_RELEASE:=1
PKG_BUILD_DIR:=$(BUILD_DIR)/hello
```

```
include $(INCLUDE_DIR)/package.mk

define Package/hello
    TITLE:=Hello World
    DESCRIPTION:=\
    A simple HELLO WORLD program
endef

define Build/Compile
    $(TARGET_CC) $(TARGET_CFLAGS) -o
    $(PKG_BUILD_DIR)/hello main.c
endef

define Package/hello/install
    $(INSTALL_DIR) $(1)/usr/bin
    $(INSTALL_BIN) $(PKG_BUILD_DIR)/hello.out $(1)/usr/
    bin/
endef

$(eval $(call BuildPackage,hello))
```

Proměnné *PKG_NAME*, *PKG_VERSION*, *PKG_RELEASE* slouží k zapsání základních informací o balíčku. Na základě těchto proměnných je generován název balíčku. Proměnná *BUILD_DIR* určuje místo, kam se bude balíček kompilovat. V sekci *define/Package hello* se nachází popis balíčku a případné závislosti. Další sekce je *Build/Compile* obsahuje skript, který zkompiluje náš program. Poslední část je *Package/hello/install*, ta určuje, jaké soubory a kam se budou kopírovat při instalaci našeho balíčku. Kompilaci spustíme příkazem:

```
make
```

Hotový balíček se nachází ve složce *bin/packages*, výsledný soubor nese tento název *hello_1.0-1_mipsel.ipk*. Náš balíček nahrajeme pomocí SCP do routerboardu:

```
scp hello_1.0-1_mipsel.ipk root@192.168.6.2:/tmp/
```

Balík se uloží v adresáři */temp* kde ho nainstalujeme následujícím příkazem:

```
ipkg install hello_1.0-1_mipsel.ipk
```

Tento postup je pouze jednou z možností tvorby vlastních balíčků. Existují další možnosti a potřeby při jejich tvorbě, automatizované stahování zdrojových kódů, následná verifikace, potřeba řešit závislosti na jiných balících a podobně. Problematika tvorby balíčků má proto komplexní charakter.

Závěr

Domnívám se, že vytyčený úkol se mi podařilo úspěšně splnit. Výsledkem práce je funkční modulární systém se správou balíků připravený na nejrůznější využití v síťovém provozu. Navíc linuxová distribuce OpenWrt i routerboardy Mikrotik nacházejí reálné využití v praxi, zejména u lokálních internetových providerů. Obecně přidaná hodnota distribuce OpenWrt spočívá také v tom, že dokáže přinést nové funkce, které dané zařízení s původním firmware nemá. S odpovídajícím hardware lze i laciný SOHO router proměnit například síťový audio přehrávač, storage server, backup server, arp proxy, telefonní ústřednu a jiné.

V první části práce byli vybráni vhodní kandidáti pro nahrazení firmware embedded Linuxem, další část se věnovala jeho praktické realizaci, počáteční konfiguraci zařízení a na závěr je zdokumentován proces tvorby vlastních softwarových balíčků.

Práce poskytuje možnost navázat v dané problematice v podobě nějaké konkrétní aplikace, kterou nenabízí dodávaný firmware. Možnost k pokračování skrývá také problematika tvorby softwarových balíčků pro distribuci OpenWrt. Ta je pro svoji obsáhlost bohužel nad rámec bakalářské práce. Nabízí nejrůznější možnosti včetně automatizace stažení zdrojových kódů, atp.

Seznámení s kategorií embedded zařízení, prohloubení znalostí o operačním systému Linux a základní průnik do problematiky nasazení OpenWrt na embedded zařízení, jako aktivních síťových prvků, pokládám za osobní přínos této práce. Věřím, že získané znalosti zúročím i později.

Bibliografie

Seznam literatury

- [1] Kolektiv autorů: *LINUX – dokumentační projekt*. Computer Press 1998 (3. aktualizované vydání), ISBN 80-7226-761-2.
- [2] *Embedded systém*. [online]. Wikipedia. 2009-27-1, [cit. 2009-04-14]. Dostupné z: <http://cs.wikipedia.org/wiki/Embedded_sys%C3%A9m>.
- [3] Štrauch, Adam: *Mikrotik: Seznámení s wi-fi krabičkou* [online]. Root.cz, 2008-11-07 [cit. 2009-04-14]. Dostupné z: <<http://www.root.cz/clanky/mikrotik-seznameni-s-wi-fi-krabickou/>>.
- [4] *License levels* [online]. Wikipedia. 2009-01-29, [cit. 2009-04-15]. Dostupné z: <http://wiki.mikrotik.com/wiki/License_levels>.
- [5] *Geode (procesor)*. [online]. Wikipedia. 2009-04-09, [cit. 2009-04-15]. Dostupné z: <[http://en.wikipedia.org/wiki/Geodezie_\(processor\)#Geode_LX](http://en.wikipedia.org/wiki/Geodezie_(processor)#Geode_LX)>.
- [6] PC Engines. *Alix3c3 system board*. [online]. 2009, [cit. 2009-02-12]. Dostupné z: <<http://www.pcengines.ch/alix3c3.htm>>.
- [7] Broadcom Corporation. BCM5354. [online]. 2008, [cit. 2009-02-12]. <<http://www.broadcom.com/products/Wireless-LAN/802.11-Wireless-LAN-Solutions/BCM5354>>.
- [8] ASUSTeK Computer Inc. WL-520gU. [online]. 2008, [cit. 2009-02-14]. Dostupné z: <http://www.asus.com/Product.aspx?P_ID=cOWUB0XOSysr4sBM>.

- [9] *OpenWrt Overview*. [online]. Wikipedia. 2009-03-29, [cit. 2009-04-16]. Dostupné z: <<http://oldwiki.openwrt.org/OpenWrtOverview.html>>.
- [10] Memory Technology device (MTD) Subsystem for Linux. [online]. 2009-01-12, [cit. 2009-02-20]. Dostupné z: <<http://www.linux-mtd.infradead.org/faq/general.html>>.
- [11] OpenWrt howto. [online]. 2009-03-29, [cit. 2009-04-21]. Dostupné z: <[http://oldwiki.openwrt.org/OpenWrtDocs/Hardware/Mikrotik/RB433.html?highlight=\(OpenWrtDocs/Hardware\)](http://oldwiki.openwrt.org/OpenWrtDocs/Hardware/Mikrotik/RB433.html?highlight=(OpenWrtDocs/Hardware))>.

Seznam tabulek a obrázků

Seznam tabulek

Tabulka 1: Přehled verzí licence RouterOs. [4].....	11
Tabulka 2: Přehled faktorů zohledněných při výběru embedded zařízení.....	17
Tabulka 3: Hlavní rozdíly mezi Block a MTD devices. [10].....	35

Seznam ilustrací

Ilustrace 1: Podoba routerboardu Mikrotik RB433AH.....	12
Ilustrace 2: Podoba embedded zařízení Alix3c3.....	14
Ilustrace 3: Podoba embedded zařízení WL-520GU.....	15
Ilustrace 4: Konfigurační prostředí buildroot.....	26
Ilustrace 5: Bootovací obrazovka Mikrotik RB433AH.....	31
Ilustrace 6: Bootup menu Mikrotik RB433ah.....	31
Ilustrace 7: Počátek úspěšného bootu po síti.....	33
Ilustrace 8: Terminál právě nastartovaného OpenWrt.....	35
Ilustrace 9: Ukázka struktury konfiguračního souboru pro ethernetové rozhraní.....	36
Ilustrace 10: Konfigurační soubor pro balíčkovací systém.....	36

Seznam použitých zkratek

CPU: Central Processing Unit

DHCP: Dynamic Host Configuration Protocol

JFFS2: Journalling Flash File system, version 2

MTD: Memory Technology Device

PCI: Peripheral Component Interconnect

PDA: Personal Digital Assistant

ROM : Read Only Memory

RPSMA: Reverse Polarity SubMiniature version A

SDK: Software Development Kit

SVN: Subversion

TCP/IP: Transmission Control Protocol / Internet Protocol

USB: Universal serial Bus

VGA: Video Graphics Array