

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PORTACE PERLU PRO OPERAČNÍ SYSTÉMY WINDOWS MOBILE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FILIP SCHAD

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PORTACE PERLU PRO OPERAČNÍ SYSTÉMY WINDOWS MOBILE

PORTING PERL ON WINDOWS MOBILE OS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FILIP SCHAD

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAROSLAV RÁB

BRNO 2011

Abstrakt

Tato bakalářská práce se zabývá portováním interpretru skriptovacího jazyka Perl na zařízení podporující operační systémy Windows Mobile. V první části se zaměřuje na představení operačního systému Windows Mobile a samotného jazyka Perl. V dalších částech pak popisuje nástroje a nutné úpravy potřebné pro úspěšný překlad. Poslední kapitola je věnována testování interpretru na cílové platformě.

Abstract

This bachelor's thesis deals with porting of scripting language Perl interpreter on Windows Mobile based devices. The first part focuses on introducing the Windows Mobile operating system and Perl itself. Next parts describes tools and necessary modifications for successful compilation. Last chapter is devoted to tests of interpreter on Windows Mobile platform.

Klíčová slova

Perl, Windows Mobile, skriptovací jazyk, smartphone, mobilní zařízení.

Keywords

Perl, Windows Mobile, scripting language, smartphone, mobile devices.

Citace

Filip Schad: Portace Perlu pro operační systémy Windows Mobile, bakalářská práce, Brno, FIT VUT v Brně, 2011

Portace Perlu pro operační systémy Windows Mobile

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rába. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Filip Schad
17. května 2011

Poděkování

Chtěl bych poděkovat svému vedoucímu práce panu Ing. Jaroslavu Rábovi za vstřícné jednání, podporu a pomoc při řešení této bakalářské práce.

© Filip Schad, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	4
2 Technologie	5
2.1 Windows Mobile	5
2.1.1 Společné rysy	5
2.1.2 Jednotlivé verze	6
2.2 Perl	9
2.2.1 Syntaxe	9
2.3 Architektura ARM	11
2.4 Knihovna CeLib	12
2.5 Knihovna W32Console	13
3 Použité nástroje	14
3.1 Microsoft Visual Studio	14
3.2 Microsoft Device Emulator	14
3.3 Nástroje pro hostitelskou platformu	14
3.4 Nástroje pro cílovou platformu	15
4 Překlad a sestavení výsledného balíčku	16
4.1 Překlad knihovny CElib	16
4.2 Překlad knihovny W32console	17
4.3 Překlad pro Windows	17
4.4 Překlad pro Windows Mobile	18
4.4.1 Makefile.ce	18
4.4.2 Změny v ostatních souborech	19
4.5 Kompletace výsledné distribuce	20
4.6 Příprava instalačního balíčku	20
5 Instalace a testy	23
5.1 Instalace Perlu na zařízení	23
5.2 Testovací rozhraní	23
5.2.1 Definice vlastních testů	24
5.3 Sada testů	24
5.4 Testy v Microsoft Device Emulator	25
5.5 Testy na hardwarových zařízeních	25
5.5.1 Testy na HTC Touch Diamond 2	26
5.5.2 Testy na Mio A701	27
5.5.3 Testy na Acer n30	27

6 Závěr	29
A Obsah CD	32

Seznam obrázků

2.1	Obrazovka dnes ve Windows Mobile 2003	6
2.2	Obrazovka dnes ve Windows Mobile 5.0	7
2.3	Domácí obrazovka ve Windows Mobile 6.5	8
2.4	Ukázka Windows Phone 7	8
3.1	Windows Mobile 6.5 Professional v prostředí Microsoft Device Emulator . .	15
4.1	Adresářová struktura potřebná pro kompilaci Perlu	16
4.2	Adresářová struktura distribuce Perlu	20
4.3	Adresářová struktura složky <i>.pl</i> v registrech	21
4.4	Adresářová struktura složky <i>perlfile</i> v registrech	21
4.5	Struktura adresáře pro vytvoření instalačního balíčku	22
5.1	Perl spuštěný v Microsoft Device Emulator	25
5.2	HTC Touch Diamond 2	26
5.3	Mio A701	27
5.4	Acer n30	28

Kapitola 1

Úvod

Stále více aplikací se přesouvá z osobních počítačů na mobilní zařízení. Umožňuje to rozmach trhu mobilních telefonů s operačními systémy a jejich relativní dostupnost. Jeden z velkých hráčů na poli operačních systémů pro tzv. „chytré telefony“ je i společnost Microsoft se svým Windows Mobile.

Programovací jazyk Perl je užitečným pomocníkem pro psaní nejrůznějších skriptů a konzolových utilit na mnoha platformách od přenosných zařízení po mainframové počítače. Avšak port tohoto interpretovaného jazyka pro Windows Mobile není dlouhodobě podporován, tudíž není k dispozici nová verze na této platformě.

Na začátku představím Windows Mobile a jazyk Perl. V následujících kapitolách se budu věnovat nástrojům a potřebným úpravám, které jsou nezbytné pro překlad interpretru tohoto jazyka. Ukážu jakým způsobem lze postupovat pro vytvoření distribuce Perlu pro operační systémy Windows Mobile. Použitá řešení se snaží být co nejuniverzálnější tak, aby mohl být postup použit pro překlad budoucích verzí. V závěru se budu věnovat testování přeloženého interpretru na virtuálním stroji a hardwarových zařízeních.

Kapitola 2

Technologie

2.1 Windows Mobile

Windows Mobile je mobilní operační systém společnosti Microsoft, vyvinutý pro použití v „chytrých telefonech“ a mobilních zařízeních.

Aktuální a poslední verze je Windows Mobile 6.5 a bude postupně nahrazen nedávno uvedeným Windows Phone 7. Je založen na jádře Windows CE 5.2 a obsahuje sadu základních aplikací vyvinutých pomocí Microsoft Windows API. Vývoj aplikací je umožněn i vývojářům třetích stran, kterým Microsoft poskytuje ucelené vývojové prostředí a SDK (Software Development Kit). Aplikace je možné stáhnout nebo zakoupit také přes *Windows Marketplace for Mobile*.

Počet uživatelů Windows Mobile v poslední době rapidně poklesl, přesto je však celosvětově patým nejpoužívanějším systémem pro mobilní zařízení (po Symbianu, BlackBerry OS, Androidu a iOS) s 5% podílem trhu. Ve spojených státech si stále drží třetí příčku v byznysové sféře (po BlackBerry OS a iOS) s tržním podílem 24% [3].

2.1.1 Společné rysy

Následující funkce a vlastnosti jsou společné pro většinu verzí:

- Obrazovka dnes, která ukazuje aktuální datum, informace o majiteli, nadcházející upomínky, e-mailové zprávy a úkoly (Ve vyšších verzích přejmenována na domácí obrazovku).
- Aplikace Office Mobile, které jsou mobilní verzí kancelářského balíku MS Office. Součástí je i MS Outlook, který slouží ke správě e-mailů a upomínek.
- Internet Explorer Mobile. Mobilní verze internetového prohlížeče firmy Microsoft.
- Klient pro PPTP VPN.
- Sdílení internetového připojení, což umožňuje sdílet připojení z mobilu pomocí USB nebo bluetooth.
- Souborový systém postavený na bázi desktopových MS Windows s podporou mnoha stejných typů souborů.
- Možnost multitaskingu.

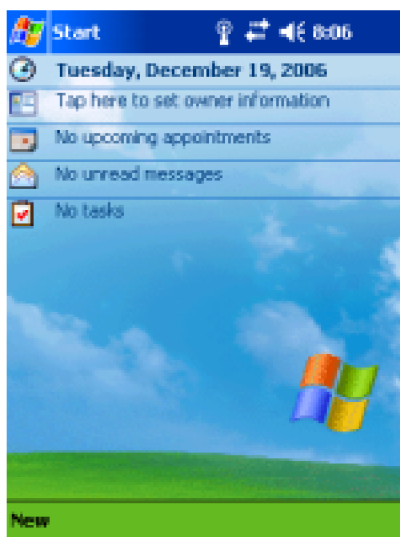
2.1.2 Jednotlivé verze

V této sekci představím několik jednotlivých verzí systému Windows Mobile, na kterých byl systém testován. Pro úplnost je doplněna informace o Windows Phone 7, kterým se tato práce přímo nezabývala.

Windows Mobile 2003

Windows Mobile 2003 (známý jako wm2003 nebo WM2003), původně známý pod kódovým označením Ozone, byl uveden 23. června 2003. Byla to zároveň první verze vydaná pod názvem Windows Mobile. Byl vydáván ve čtyřech verzích: Windows Mobile 2003 for Pocket PC Premium Edition, Windows Mobile 2003 for Pocket PC Professional Edition, Windows Mobile 2003 for Smartphone, Windows Mobile 2003 for Pocket PC Phone Edition. Verze uvedená jako poslední byla speciálně navržena pro kapesní počítače s integrovanými telefonními funkcemi. Windows Mobile 2003 používá jádro Windows CE 4.20 [3].

Na následujícím obrázku 2.1 je ukázka obrazovky dnes v systému Windows Mobile 2003.



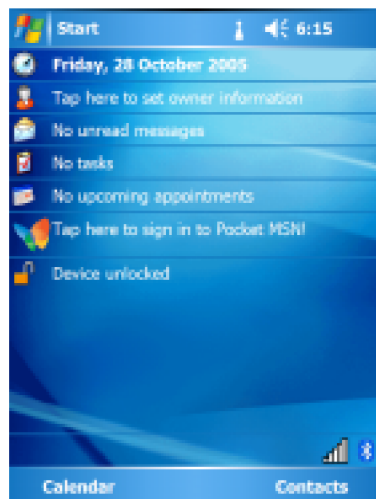
Obrázek 2.1: Obrazovka dnes ve Windows Mobile 2003

Windows Mobile 5

Windows Mobile 5.0, původně známý pod kódovým označením Magneto byl poprvé představen na Microsoft Mobile and Embedded Developers Conference v roce 2005 v Las Vegas 9. května 2005. Zákaznická podpora pro tento systém trvala do 12. října 2010 a rozšířená bude pokračovat až do 13. října 2015. První zařízením s tímto operačním systémem byl Dell Axim x51 [3].

Windows Mobile 6

Windows Mobile 6, formálně známý pod kódovým označením Crossbow, byl uveden 12. února 2007 na 3GSM World Congress 2007. Přišel ve třech různých verzích [3]:



Obrázek 2.2: Obrazovka dnes ve Windows Mobile 5.0

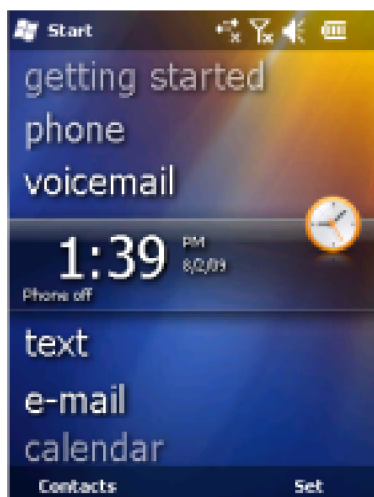
- Standard: pro telefony bez dotykové obrazovky
- Professional: pro mobilní zařízení s telefonním modulem
- Classic: pro mobilní zařízení bez telefonního modulu

Windows Mobile 6 je postaven na jádru Windows CE 5.0 (později 5.2) a je silně provázán se službami Windows Live a Exchange 2007. Po stránce estetiky byl tento systém slazen s tehdy nově uvedeným systémem Windows Vista. Po stránce funkční je velmi podobný Windows Mobile 5, ale vyznačuje se vyšší stabilitou.

V rámci verze 6 vyšli ještě dvě novější verze 6.1 a 6.5.

Verze 6.1 uvedená 1. dubna 2008 přišla s přepracovanou domácí obrazovkou, mobilní verzí aplikace OneNote, sdružováním SMS do vláken a několika dalšími dílčími vylepšeními. Díky vylepšení protokolu ActiveSync dosáhl výrazně větší výdrže baterie u většiny zařízení.

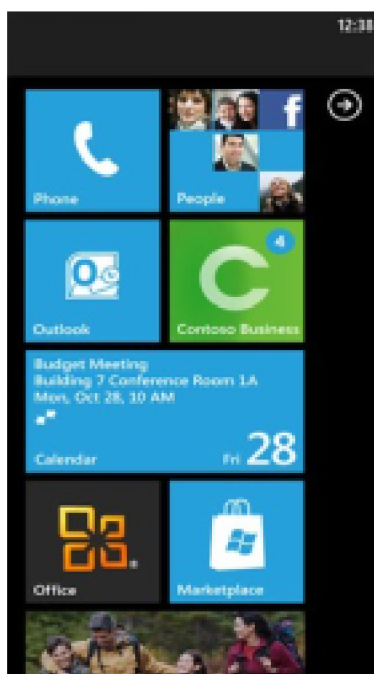
Verze 6.5 je považována pouze za aktualizaci verze 6.1 a měla být dočasným přechodným řešením do uvedení Windows Phone 7. Dále rozšiřuje přizpůsobení ovládání prsty a obsahuje novou verzi internetového prohlížeče Internet Explorer [3].



Obrázek 2.3: Domácí obrazovka ve Windows Mobile 6.5

Windows Phone 7

Windows Phone 7 se začal v Evropě, Singapuru a Austrálii prodávat 21. října 2010. V USA a Kanadě potom 8. listopadu. Zbytek světa by měl následovat během roku 2011. Microsoft se rozhodl vytvořit zcela novou platformu mobilních operačních systémů, proto většina telefonů, které v současné době běží na Windows Mobile 6.x, nepůjdou upgradovat na verzi Windows Phone 7 [3][12].



Obrázek 2.4: Ukázka Windows Phone 7

2.2 Perl

Kniha Beginning Perl [7] popisuje tento programovací jazyk následujícím způsobem:

Perl je vysokoúrovňový, víceúčelový, interpretovaný, dynamický programovací jazyk. Původně vyvinutý Larry Wallem v roce 1987 jako víceúčelový unixový skriptovací jazyk pro zjednodušení zpracování textu. Od té doby, prošel mnoha změnami a stal se široce populární mezi programátory. Larry Wall nadále dohlíží na vývoj jádra jazyka a jeho nadcházející verzi Perl 6. Půjčuje si prvky z jiných programovacích jazyků včetně C, unixového shellu (sh), AWK, a sedu. Perl poskytuje výkonné zpracování textu bez omezení délky dat, jak tomu bývá u některých současných unixových nástrojů. Umožňuje snadnou manipulaci s textovými soubory. Perl získal větší popularitu na začátku devadesátých let jako jazyk pro psaní CGI skriptů, z části díky jeho schopnostem v textové analýze.

Kromě CGI se Perl používá pro grafické programování, správu systému, síťové programování, finance, bioinformatiky a další aplikace. Perlu se přezdívá „švýcarská vojenská řetězová pila programovacích jazyků“ kvůli jeho flexibilitě a výkonu.

Perl je jeden z nejvíce portovaných programovacích jazyků. Může být přeložen na více než sedmdesátí druhých operačních systémů a jeho zkompileovaná binární verze je dostupná pro všechny nejrozšířenější platformy.

Perl je šířen pod dvěma licencemi: Artistic Licence a GNU/GPL. To umožňuje využívat ho i pro komerční účely. Jazyk má velkou základnu příznivců. Existuje proto nepřehledné množství konferencí, IRC chatů, news a dalších komunikačních kanálů, kde je možné spolupracovat a radit se s lidmi, kteří Perl používají. Zdrojový kód, dokumentace i návody lze najít na domovské stránce tohoto projektu [4].

2.2.1 Syntaxe

Na následujících řádcích se věnuji hlavním vlastnostem a syntaxi tohoto programovacího jazyka. Perl je známý především kvalitní podporou regulárních výrazů a mottem: „There’s more than one way to do it“. Většina konstrukcí jazyka je možné zapsat podobným způsobem jako v jazyce C, často je možné použít i nějakou speciální konstrukci navrženou v Perlu. Zmiňuji zde proto hlavně speciality a odlišnosti, které nejsou zcela běžné v jiných programovacích jazycích.

Poznámky

Poznámka je uvozena znakem # („hash“). Cokoliv za tímto znakem je až dokonce řádku ignorováno. Perl umožňuje pouze jednořádkové komentáře. Vývojáři Perlu tím chtěli docílit, aby programátoři psali kód, a ne poznámky. Tento přístup je silně diskutabilní. Někdo tento přístup považuje za přínos, jiný za velkou nevýhodu.

Proměnné

Proměnná v Perlu je vždy uvozena speciálním znakem, který značí o jaký typ proměnné se jedná. Dále musí splňovat pravidla pro pojmenování. Proměnná může mít jeden až 255 znaků složených z písmen anglické abecedy, čísel a podtržítka, přičemž název nesmí začínat číslem. Perl rozlišuje malá a velká písmena v názvu proměnné. To znamená, že *\$promenna* a *\$Promenna* jsou dva různé názvy. Protože je Perl dynamicky typovaný, nemusíte dopředu deklarovat typ proměnné, interpret si jí určí sám podle jejího obsahu. Tento přístup má své výhody v rychlosti a krátkosti zápisu. Může se však stát, že si do proměnné omylem

uložíte jiný typ, než jste zamýšleli. Jméno proměnné si lze deklarovat pomocí klíčového slova *my*, není to ovšem vyžadováno, pokud nepoužíváte rozšiřující balík *strict*. Existuje i možnost deklarace globální proměnné pomocí klíčového slova *our*. Proměnná může být definována v jakékoliv hloubce zanoření. Jedinou podmínkou zůstává její deklarace před prvním použitím. V tabulce 2.1 je uveden seznam typů proměnných se kterými se můžete v Perlu setkat. Tato tabulka byla převzata z knihy Programming Perl [15].

Typ proměnné	Uvozující znak	Příklad	Použití pro:
Skalární	\$	\$skalar	Jednoduchá hodnota (číslo nebo řetězec)
Pole	@	@pole	Seznam hodnot klíčovaný číslem
Hash	%	%hash	Asociativní pole hodnot
Subrutina(funkce)	&	&funkce	Volatelný kus kódu
Typeglob	*	*odkaz	Cokoliv se jménem odkaz

Tabulka 2.1: Typy proměnných v Perlu

Speciální proměnné

Perl obsahuje desítky speciálních proměnných, které můžete najít v dokumentaci. Nejpoužívanější speciální proměnné v Perlu jsou `$_` a `@_`. Tyto proměnné mění svůj obsah podle kontextu. Díky těmto vlastnostem můžete používat funkce bez předání argumentu. Pokud není argument funkce uveden, interpret implicitně předpokládá použití výchozí proměnné. Zde je ukázka možného použití těchto dvou proměnných:

```
$_ = "Hello world!";
@_ = ("a", "b");
print;                #vypíše Hello World
&tiskni;              #volání funkce tiskni vypíše ab
sub tiskni{           #definice funkce
    foreach (@_){ print; }
}
```

Regulární výrazy

Jedná se o jednu z nejmocnějších vlastností Perlu. Protože původně vznikl jako nástroj pro práci s textem, obsahuje přímo ve svém jádře možnost práce s regulárními výrazy. Tuto vlastnost si vypůjčil od unixových nástrojů jako je *grep*, *awk*, či *sed*. Díky této vlastnosti si získal velkou oblibu mezi programátory. Těžko byste hledali jazyk, který podporuje regulární výrazy komplexnějším způsobem než Perl. Spojení skriptovacího jazyka a podpory regulárních výrazů dělá z Perlu opravdu mocný a užitečný nástroj každého správce a programátora. Následující ukázka předvádí jeden ze způsobů, jak se dá s regulárními výrazy v Perlu pracovat:

```
@names = qw (Alice Bob Camilla Dave Eve);
# Cyklus vytiskne všechna jména která obsahují malé nebo velké písmeno a.
# V tomto případě vytiskne: „Alice, Camilla, Dave, “
foreach (@names) {
    print $_ . ", " if /[Aa]/;
```

}

Operátory

U binárních operátorů má Perl kromě klasických operátorů sčítání(+), odčítání(-), dělení(/) a modulo(%) i operátor pro mocninu, který se značí dvěma hvězdičkami(**).

Obsahuje také 2 operátory pro práci s řetězci. Operátor pro konkatenaci řetězců(.) a operátor pro násobení řetězců, který se zapisuje jako malé písmeno x. Příklad použití násobení řetězce:

```
$a = "ab";  
$b = 5;  
print $a x $b #výpíše abababab
```

Dalším druhem jsou operátory pro porovnání. Perl má dvě sady těchto operátorů. Jednu pro porovnávání čísel a druhou pro řetězce. Následující tabulka 2.2 ukazuje chování při použití operátoru na proměnné \$a a \$b.

Čísla	Řetězce	Návratová hodnota
==	eq	Vrací true pokud \$a je rovno \$b
!=	ne	Vrací true pokud \$a není rovno \$b
<	lt	Vrací true pokud \$a je menší než \$b
>	gt	Vrací true pokud \$a je větší než \$b
<=	le	Vrací true pokud \$a není větší než \$b
<=>	cmp	Vrací 0 pokud jsou stejné, 1 pokud je \$a větší, -1 pokud je větší \$b

Tabulka 2.2: Typy operátorů pro porovnání

Posledním druhem operátorů, o kterém se zde zmíním jsou operátory pro testování souborů. Tyto operátory umožňují zjišťovat zda jsou nastaveny některé atributy souboru dřív než s ním začnete pracovat. Nejzákladnější atribut je samozřejmě, jestli soubor existuje. Díky tomu se dá předejít nedopatření, kdy daný soubor otevřete jako nový a přepíšete tím vše, co už tam bylo. Tabulka 2.3 ukazuje některé z těchto operátorů:

Příklad	Druh testu	Návratová hodnota
-e \$s	existuje	Vrací true pokud \$s existuje
-r \$s	lze číst	Vrací true pokud lze z \$s číst
-w \$s	lze zapisovat	Vrací true pokud lze do \$s zapisovat
-d \$s	adresář	Vrací true pokud je \$s adresářem
-f \$s	soubor	Vrací true pokud je \$s soubor
-T \$s	textový soubor	Vrací true pokud je \$s textový soubor

Tabulka 2.3: Typy operátorů pro testování souborů

2.3 Architektura ARM

Okolo 98% všech vyrobených telefonů používá nejméně jeden chip založený na architektuře ARM. Tím je tato architektura v současnosti jednoznačně nejpoužívanější u tohoto typu

zařízení. Nepoužívá se však pouze u mobilních telefonů, ale také u dalších zařízení jako jsou set-top boxy, digitální kamery, tiskárny, televize a další [11].

ARM procesory využívají architekturu s redukovanou instrukční sadou, pro kterou se používá zkratka RISC (z anglického **R**educed **I**nstruction **S**et **C**omputer). Jako první se vývojem této architektury začaly zabývat univerzity ve Standfordu a Berkeley počátkem osmdesátých let. První ARM procesor byl vyvinut společností Acorn Computers Limited v Cambridge ve Velké Británii v letech 1983 až 1985. Byl to první procesor určený pro komerční použití postavený na architektuře RISC. V roce 1990 vznikla společnost ARM Limited, která se zabývá vývojem technologie ARM. Tato společnost vlastní veškerá intelektuální práva k architektuře ARM, ale nezabývá se jejich výrobou, pouze prodává licence ostatním společnostem mezi které patří přední světové společnosti na výrobu procesorů a polovodičových součástek [8].

Charakteristika procesoru ARM:

- 32bitová vnitřní architektura
- 32bitová datová sběrnice s propustností 32 MB/s
- 26bitová adresová sběrnice (dostupný lineární prostor 64MiB)
- 25 vnitřních 32bitových registrů
- přístup do paměti pouze instrukcemi Load a Store
- částečné překrývání vnitřních registrů
- možnost podmíněného vykonání instrukcí
- možnost připojení standardních pamětí DRAM
- jednoduchý a výkonný instrukční soubor, jednoduše využitelné kompilátory vyšších programovacích jazyků

2.4 Knihovna CeLib

Jedná se o knihovnu od Rainera Keuchela, která implementuje crt, stdio, unixové a Windows ansi funkce pro Windows Mobile. Je velmi užitečná pro portování programů z Windows i unixových systémů na tuto mobilní platformu. Poslední vydaná verze je 3.11. Její zdrojové kódy i zkompileovaná verze pro nejběžnější platformy je dostupná na webu sourceforge.net [9].

Zkompileované verze jsou velmi staré. Poslední verze knihovny byla vytvořena v roce 2002. Proto se později v této práci v sekci 4.1 budu věnovat i kompilaci této knihovny. Na příloženém CD je v instalačních balíčcích použita její nově přeložená verze. Při testování jsem zjistil, že může být použita i její původní verze, která je přímo v archivu *celib-palm-3.0*. Doporučuji však použít verzi novější.

Archiv obsahuje zkompileované knihovny pro následující druhy procesorů:

- arm
- mips
- sh3

- sh4
- x86

2.5 Knihovna W32Console

Další knihovna z dílny Rainera Keuchela. Tentokrát se jedná o jednoduchou konzoli (shell) pro Windows Mobile. Její rozhraní může sloužit dalším projektům pro vstup a výstup nebo se dá používat samostatně. Zdrojové kódy poslední verze 1.3 jsou volně dostupné ke stažení na webu sourceforge.net [10].

Archiv kromě samotných zdrojových souborů obsahuje i objektové soubory zkompilevané pro jednotlivé platformy. Výčet druhů procesorů je následující:

- arm
- mips
- sh3
- sh4
- x86

Stejně jako předchozí knihovna CeLib je i tato velmi stará, také z roku 2002. Proto se v rámci překladu Perlu věnuji i překladu této knihovny v sekci 4.2.

Kapitola 3

Použité nástroje

Pro překlad bylo použito Microsoft Visual Studio 2008 Professional Edition, které obsahuje kompilátor, linker i většinu potřebných knihoven jak pro překlad samotného Perlu na platformě x86 ve Windows, tak pro následný překlad na mobilní platformu ARM.

Před samotným testováním na fyzickém hardware byl interpret testován na virtuálním stroji pomocí nástroje Microsoft Device Emulator, který umožňuje simulovat reálný hardware. Cílem bylo předejít případným problémům s poškozením systému na reálných zařízeních.

3.1 Microsoft Visual Studio

Microsoft Visual Studio je integrované vývojové prostředí od společnosti Microsoft. Může být použito k vývoji grafických i konzolových aplikací, webových stránek, webových aplikací nebo služeb jak pro osobní počítače, tak i pro mobilní zařízení. Podporované platformy jsou Microsoft Silverlight, .NET Framework, .NET Compact Framework, Microsoft Windows, Windows Mobile a Windows CE. Jeho součástí je i MS SQL Server Developer Edition.

3.2 Microsoft Device Emulator

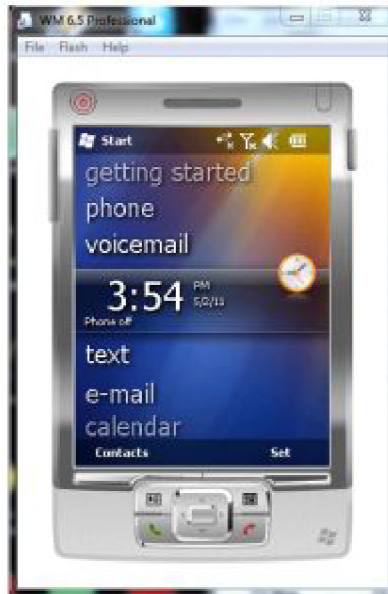
Microsoft Device Emulator je nástroj sloužící k napodobení chování skutečné platformy pro Windows Mobile. Poskytuje virtuální hardware, na kterém je možno testovat aplikace. Obsahuje velkou škálu nastavení tak, aby si každý vývojář tento virtuální stroj mohl přizpůsobit co nejvíce svým potřebám. Mezi vlastnosti, které jdou nastavit, patří například rozlišení obrazovky, orientace displeje, emulace sériových portů, emulace paměťové karty a mnoho dalších. Díky tomu může programátor otestovat jakoukoli aplikaci bez obav, že dojde k poškození systému na skutečném hardware.

Na obrázku 3.1 je ukázka spuštěného emulátoru se systémem Windows Mobile 6.5.

3.3 Nástroje pro hostitelskou platformu

Pro překlad hostitelské části bylo využito následujících programů se sady MS Visual Studio 2008:

- Program nmake (VC\bin\nmake.exe)
- Kompilátor cl pro platformu x86 (VC\bin\cl.exe)



Obrázek 3.1: Windows Mobile 6.5 Professional v prostředí Microsoft Device Emulator

- Linker link pro platformu x86(VC\bin\link.exe)

Dále je nutné nainstalovat soubor knihoven a hlavičkových souborů pro vývoj na platformě Windows (tzv. SDK – Software Development Kit).

3.4 Nástroje pro cílovou platformu

Pro překlad hostitelské části a sestavení výsledné distribuce bylo využito následujících programů a knihoven z MS Visual Studio:

- Program nmake(VC\bin\nmake.exe)
- Kompilátor cl pro platformu ARM(VC\ce\bin\x86_arm\cl.exe)
- Linker link pro platformu ARM(VC\ce\bin\x86_arm\link.exe)
- Knihovny a hlavičkové soubory pro platformu ARM.
- Nástroj *cabwiz* pro vytvoření instalačního balíčku pro platformu Windows Mobile.

Další knihovny nutné pro překlad na Windows Mobile:

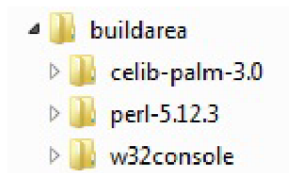
- W32console od Rainera Keuchela
- CELib od Rainera Keuchela

Kapitola 4

Překlad a sestavení výsledného balíčku

V době vytváření této práce byl nejnovější uvolněný zdrojový kód Perlu verze 5.12.3. Ten byl také použit pro překlad. Poslední překládaná verze pro Windows Mobile byla 5.8.2, pro kterou byl sepsán návod na přeložení [6]. Tento návod sloužil jako odrazový můstek k vytvoření funkčního překladu. Není však kompletní, navíc je silně zastaralý, takže bylo potřeba na velkou část překladu přijít, nejčastěji metodou pokus, omyl. To je sice zdlouhavý způsob, vzhledem k absenci jakéhokoliv kvalitního manuálu, však jediný možný.

Pro překlad jsem si vytvořil složku, kam jsem zkopíroval veškeré potřebné zdrojové soubory. Vznikla tím kolekce zdrojových souborů potřebných pro kompilaci. Tato struktura je zobrazena na obrázku 4.1. Adresář *celib-palm-3.0* obsahuje zdrojové soubory knihovny CElib, *perl-5.12.3* zdrojové soubory Perlu a *w32console* zdrojové soubory knihovny W32console. Archivy s těmito soubory jsou přítomny na přiloženém CD.



Obrázek 4.1: Adresářová struktura potřebná pro kompilaci Perlu

Samotný překlad Perlu pro Windows Mobile se skládá z několika dílčích kroků, které jsou pro přehlednost rozděleny a blíže popsány v jednotlivých sekcích níže.

4.1 Překlad knihovny CElib

Archiv knihovny CElib obsahuje již zkompileovanou dynamickou knihovnu. Tento překlad byl vytvořen v roce 2002, primárně pro operační systém Windows CE 3.0. Rozhodl jsem se proto knihovnu překompilovat pro zajištění funkčnosti na novějších zařízeních. Soubor *Makefile.ce* určený pro kompilaci byl nefunkční, vytvořil jsem proto nový, který zakládá na staré verzi. Ve složce *changed_files* na přiloženém CD je soubor *Makefile-celib*. Ten stačí překopírovat do kořenové složky zdrojových souborů knihovny. Dále je v něm potřeba nastavit několik proměnných nutných pro překlad:

- CECONSOLEDIR – umístění zdrojových souborů knihovny a aplikace W32console
- CEPATH – cesta k adresáři s kompilátorem a linkerem pro výslednou platformu (např. C:\Program Files\Microsoft Visual Studio 9.0\VC\ce\bin\x86_arm)
- INCLUDE – adresáře s hlavičkovými soubory
- LIB – adresáře s knihovnami

Dále bylo třeba upravit hlavičkový soubor *cewin32.h*. Ten je součástí knihovny CELib a nachází se v podadresáři *inc*. V tomto souboru je potřeba zakomentovat definici struktury *_ABC* na řádcích 108 až 112. Kód koliduje se stejnojmennou funkcí, která je již definována v systémových knihovnách pro Windows Mobile.

Pokud je vše správně nastaveno stačí v příkazovém řádku spustit příkaz *nmake -f Makefile-celib* a knihovna se zkompiluje. Přeložená knihovna se po překladu nachází v podadresáři *wince-arm-pocket-wce300-release*.

4.2 Překlad knihovny W32console

Archiv knihovny W32console obsahuje již objektové soubory vytvořené kompilátorem. Tento překlad byl vytvořen v roce 2002, primárně pro operační systém Windows CE 3.0. Rozhodl jsem se proto soubory překompilovat pro zajištění funkčnosti na novějších zařízeních. Soubor *Makefile.ce* určený pro kompilaci byl nefunkční, vytvořil jsem proto nový, který je založen na původní verzi. Ve složce *changed_files* na přiloženém CD je soubor *Makefile-w32console*. Ten stačí přepokopírovat do kořenové složky zdrojových souborů knihovny. Dále je v něm potřeba nastavit několik proměnných nutných pro překlad:

- CELIBDIR – umístění zdrojových souborů knihovny CeLib
- CEPATH – cesta k adresáři s kompilátorem a linkerem pro výslednou platformu (např. C:\Program Files\Microsoft Visual Studio 9.0\VC\ce\bin\x86_arm)
- INCLUDE – adresáře s hlavičkovými soubory
- LIB – adresáře s knihovnami

Pokud je vše správně nastaveno stačí v příkazovém řádku spustit příkaz *nmake -f Makefile-w32console* a soubory se zkompilují. Přeložené objektové soubory se po překladu nachází v podadresáři *wince-arm-pocket-wce300*. Kromě nich vznikne ještě aplikace *w32console*, která se dá používat jako samostatná aplikace na zařízeních s Windows Mobile. Jedná se o jednoduchý příkazový řádek (shell). Pro samotný běh Perlu však tato aplikace není nezbytná.

4.3 Překlad pro Windows

Dříve než se Perl začne překládat pro cílovou platformu, je nutné zkompilovat Perl pro Windows a vytvořit tak potřebné binární soubory a knihovny, které se dále využívají pro překlad na mobilní platformu. Jelikož je port Perlu pro operační systém Microsoft Windows udržovaný, s touto částí nejsou žádné výrazné problémy.

V sobou *Makefile*, v adresáři *win32* stačí vybrat správnou verzi překladače. Tato změna se provádí na řádcích 98 až 117. Standardně je vybráno Microsoft Visual Studio 6

(*CCTYPE=60*). Jelikož jsem používal verzi 2008 Professional, bylo potřeba zakomentovat výchozí výběr a odkomentovat řádek 117 (*CCTYPE=90*).

4.4 Překlad pro Windows Mobile

Následný překlad na cílovou platformu je jádro celé této práce, protože bylo potřeba zjistit jaké soubory, jakým způsobem modifikovat tak, aby bylo možné interpret přeložit na platformu Windows Mobile.

Postup překladu se skládal z mnoha iterací, než bylo docíleno uspokojivého výsledku. S každou úpravou byl potřeba provést nový překlad a zjistit, kde se vyskytuje další chyba. Tímto způsobem byl upravován zdrojový kód Perlu, přidávány závislosti na nové knihovny a další kroky, dokud nebylo dosaženo funkčnosti interpretu na cílové platformě.

Výsledkem těchto úprav jsou změněné zdrojové soubory. Ty jsou ve složce *changed_files* na příloženém CD. Po překladu na Windows popsáném v kapitole 4.3, je potřeba nahradit soubory ve zdrojovém adresáři těmito soubory, aby překlad fungoval. Následující podsekcce přibližují jakým způsobem byly upravené jednotlivé soubory. Je u nich napsáno, kde se vyskytují, aby mohly být nahrazeny v případě překladu novější verze nebo dalších rozšíření této práce.

Po nahrazení souborů a správném nastavení proměnných v souboru *Makefile.ce* stačí v příkazové řádce spustit příkaz *nmake -f Makefile.ce* a spustit tím kompilaci perlu pro Windows Mobile. Po překladu se nachází soubory *perl.exe* a *perl512.dll*, které jsou výsledkem kompilace, ve složce *win32\wince-arm-pocket-wce300*.

4.4.1 Makefile.ce

Z důvodu dlouhodobé nečinnosti v rámci překladu Perlu na platformy Windows Mobile, je *makefile* pro vytvoření distribuce značně zastaralý a nefunkční. Proto jsem se rozhodl vytvořit zcela nový, který ovšem čerpá jednotlivé části z původního *Makefile.ce*. Pomíjí však některé části a nastavení, která buď nejsou potřebná nebo nefungují. Původní *Makefile.ce* je ve složce *win32* v kořenovém adresáři Perlu. Základem je správné nastavení všech proměnných pro překlad. Z důvodu velkého počtu závislostí a proměnných v rámci *makefile* zde uvedu jen nastavení nejdůležitějších proměnných, které je potřeba provést pro překlad na různých strojích, případně novější verze.

- *PV* – verze knihovny
- *INST_VER* – verze Perlu, který je překládán
- *CEPATH* – cesta k adresáři s kompilátorem a linkerem pro výslednou platformu (např. *C:\Program Files\Microsoft Visual Studio 9.0\VC\ce\bin\x86_arm*)
- *CELIBDIR* – umístění zdrojových souborů knihovny *CeLib*
- *CECONSOLEDIR* – umístění zdrojových souborů knihovny a aplikace *W32console*
- *MSVCDIR* – cesta k adresáři *VC*, který je součástí Visual Studia (např. *C:\Program Files\Microsoft Visual Studio 9.0\VC*)
- *INCLUDE* – adresáře s hlavičkovými soubory
- *LIB* – adresáře s knihovnami

4.4.2 Změny v ostatních souborech

Kromě samotného *Makefile.ce* bylo zapotřebí upravit ještě několik dalších souborů.

wince.c

Soubor se nachází ve zdrojových kódech Perlu v podadresáři *win32*. Změna je potřeba provést ve funkci *win32_get_xlib*. Musí se zde zakomentovat deklarace dvou proměnných *len* a *newsiz* na řádce 265. Proměnná *len* koliduje s parametrem funkce a proměnnou *newsiz* funkce vůbec nepoužívá. Dále je nutné přepsat řádek 296 z **len = SvCUR(sv1);* na *SvCUR(sv1);*, protože proměnná *len* je předávána funkci jako konstanta. Poslední změnou v tomto souboru je zakomentování nebo smazání funkce *isnan*, která se nachází na řádcích 2740 až 2744. Jedná se o nepovolenou redefinici.

perllib.c

Soubor je uložen v podadresáři *win32* ve zdrojových kódech Perlu. Ve funkci *RunPerl* je podmínka, která která rozděluje verze na Windows 2000 a novější v prvním případě, starší v druhém. Dělá tak na základně majoritního čísla verze. Vývojová řada Windows Mobile se však liší. Proto je potřeba zakomentovat první část podmínky tak, aby se vždy provedla pouze větev *else*. Znamená to zakomentování řádků 223 až 228 a řádku 232. Tím bylo docíleno korektního chování na Windows Mobile.

time64.config.h

Tento soubor je přímo v kořenovém adresáři zdrojových souborů Perlu. Definuje použití 64bitových časových značek. Toto opatření implementovali vývojáři Perlu kvůli používání roku vyššího než 2038. Windows Mobile však nemá knihovny pro podporu 64bitových datových formátů, a proto je potřeba toto chování vypnout. Na řádce 29 je potřeba zakomentovat makro *#define USE_TM64*. Dále byly nahrazeny makra na řádcích 80 až 83 následujícími:

```
#define SYSTEM_LOCALTIME_MAX 2147483647
#define SYSTEM_LOCALTIME_MIN 0
#define SYSTEM_GMTIME_MAX 2147483647
#define SYSTEM_GMTIME_MIN 0

#ifdef UNDER_CE
#define GMTIME_MAX 2147483647
#define GMTIME_MIN 0
#define LOCALTIME_MAX 2147483647
#define LOCALTIME_MIN 0
#endif
```

time64.c

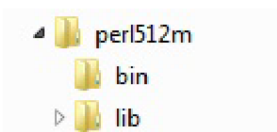
Kód se nachází přímo v kořenovém adresáři Perlu. Zde je potřeba zakomentovat dvě podmínky ve funkci *S_localtime64_r*. Jedná se o řádky 479 až 490 a 508 až 511.

perldll.def

Tento soubor je nutné nahrát do složky win32 ve zdrojových kódech Perlu. Soubor obsahuje definici exportovaných funkcí pro správné vytvoření hlavní knihovny interpretu. Bez tohoto definičního souboru by linkování knihovny *perl512.dll* skončilo chybou, případně by byla knihovna nepoužitelná.

4.5 Kompletace výsledné distribuce

Po zkompilování interpretu bylo potřeba vytvořit distribuční balík, který by byl jednoduše přenositelný a neobsahoval zbytečné soubory navíc, které by zabíraly místo na paměťové kartě telefonu. Distribuce má adresářovou strukturu znázorněnou na obrázku 4.2.



Obrázek 4.2: Adresářová struktura distribuce Perlu

Adresář *Per512m* na obrázku 4.2 je kořenovým adresářem distribuce Perlu pro Windows Mobile. Podadresář *bin* obsahuje 2 soubory – *perl.exe* a *perl512.dll*. Ty jsou výsledkem úspěšného překladu. Nachází se ve složce *win32\wince-arm-pocket-wce300* od kořenového adresáře Perlu, ve kterém byl proveden překlad. Druhý podadresář *lib* vznikl zkopírováním adresáře z balíku zdrojových kódů Perlu. Jelikož se při kompilaci do adresáře *lib* zasahuje, adresář byl zkopírován z původního archivu, tak aby neobsahoval zbytečné soubory a výsledky testů prováděné při kompilaci.

4.6 Příprava instalačního balíčku

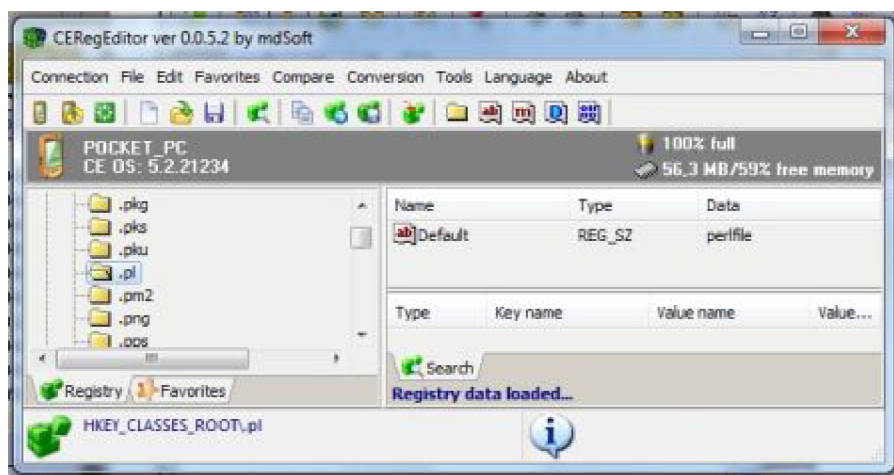
Při tvorbě instalačního balíčku jsem narazil na limity Windows Mobile. Pokud jsem do instalačního balíčku zahrnul kromě nastavení i samotný Perl, nebylo možné instalaci dokončit. Vždy po nějaké době selhala. Bylo to způsobeno množstvím souborů, které distribuce obsahuje. Při nižším počtu instalace proběhla v pořádku, avšak instalace nebyla kompletní. Proto jsem se rozhodl vytvořit instalační balíček pouze pro nastavení registrů a zkopírování knihovny celib do Windows. Distribuci Perlu jsem ponechal jako zip archiv, který je nutno zkopírovat na datovou kartu telefonu.

Při testování testování na reálných zařízeních jsem však zjistil, že chyba se týkala pouze virtuálního prostředí Device Emulator. Nakonec vznikly tedy 2 možnosti jak Perl na zařízení nainstalovat. První je kompletní instalace zapomocí instalačního balíčku *perl512m.CAB*. Druhou možností zůstává nainstalovat pouze nastavení pomocí souboru *perl-settings.CAB* a rozbalení archivu *perl512m.zip* na datovou kartu telefonu. Všechny tyto soubory se nachází na příloženém CD.

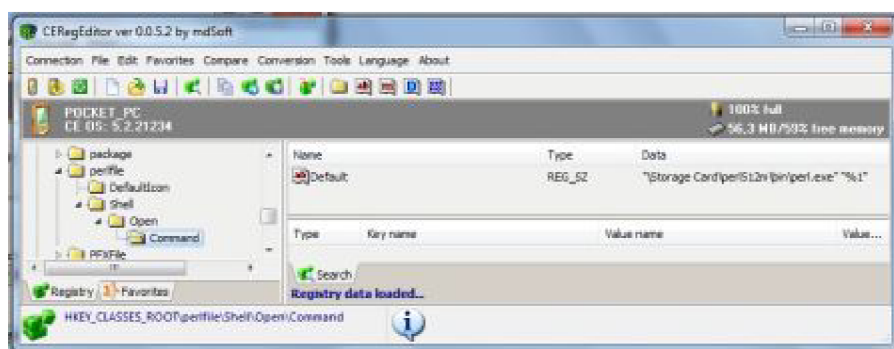
Pro vytvoření instalačního balíčku byl použit nástroj *cabwiz*, který je součástí Visual Studia. Následující kroky popisují způsob vytvoření instalačního balíčku. Při vytváření instalačního balíčku jsem čerpal z kvalitně zpracovaného návodu na serveru ce4you.cz [2].

V první řadě je zapotřebí vytvořit informační soubor s příponou *inf*. Ten obsahuje informace o jednotlivých souborech pro zkopírování, přidání klíčů do registrů, případně vytvoření zástupců v menu telefonu.

Správné nastavení registrů umožňuje spustět skripty napsané v Perlu jednoduše poklepnutím na daný skript. V připraveném instalačním balíčku na přiloženém CD jsou položky registru nastaveny pro výchozí cestu k binárním souborům Perlu (`\Storage Card\perl512m`). Pokud jste se rozhodli zkopírovat Perl na jiné umístění, musíte, pro korektní funkčnost nastavení, změnit i položky v registrech. V registrech v adresáři *HKCR* se vytvoří 2 nové složky: *.pl* a *perlfile*. První definuje příponu pro kterou platí pravidla nastavená ve složce *perlfile*. Díky tomu můžete do registrů přidávat i další přípony, které chcete používat pro skripty napsané v Perlu (například *.plx*, *.pm*, *ap.*). Složka *perlfile* obsahuje defaultní klíč s popisem typu souboru. V našem případě je hodnota *Perl Script*. Dále obsahuje další dvě podsložky *DefaultIcon*, kde je definována cesta k ikoně zobrazované u souborů s danou příponou a podsložku *Shell*. V té je definováno chování při spuštění daného souboru. To je nastaveno na spuštění Perlu s parametrem souboru, na který bylo kliknuto. Pro názornost uvádím obázky 4.3 a 4.4, na kterých je vidět adresářová struktura v registrech na jednom z testovaných telefonů.

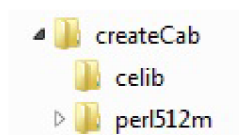


Obrázek 4.3: Adresářová struktura složky *.pl* v registrech



Obrázek 4.4: Adresářová struktura složky *perlfile* v registrech

Posledním krokem je definování souborů, které se mají do instalace zahrnout a kam se mají nainstalovat. Jelikož výsledná distribuce obsahuje 282 souborů v 39 složkách, bylo by ruční psaní takového informačního souboru opravdu zdlouhavé a při změnách by bylo potřeba náročně celý soubor přepisovat. Proto jsem za tímto účelem vytvořil jednoduchý skript *createInf.pl* napsaný v Perlu, který tento soubor generuje. Soubor se nachází na přiloženém CD. Skript pracuje následujícím způsobem. Na obrázku 4.5 je struktura adresáře *createCab*. V něm je nahraný soubor *createInf.pl*. Adresář *celib* obsahuje soubor *celib.dll* vzniklý po kompilaci knihovny CeLib a *perl512m* je distribuce Perlu pro Windows Mobile. V tuto chvíli stačí spustit skript s argumentem *perl512m* a skript projde všechny složky adresáře *perl512m*. Zjistí si tím data potřebná pro vytvoření informačního souboru. Ten vypíše na standardní výstup. Výstup stačí přeměřovat do souboru, například *perl512.inf*. Tím je informační soubor hotový. Při změnách v distribuci Perlu potom stačí do složky *createCab* nahrát změněné soubory a znovu vytvořit instalační balík nástrojem *cabwiz*. Pokud dojde k přidávání nebo mazání souborů, či složek, musí se informační soubor znovu přegenerovat spuštěním skriptu *createInf.pl* s argumentem *perl512m*.



Obrázek 4.5: Struktura adresáře pro vytvoření instalačního balíčku

Posledním krokem je vytvoření instalačního balíčku pomocí nástroje *cabwiz*. V případě, že je informační soubor správně vytvořen stačí spustit nástroj *cabwiz*, kterému dáte jako první parametr informační soubor. Ve složce, kde je informační soubor uložen vám tím vznikne stejnojmenný soubor s příponou *CAB*. Při vytváření balíčku je dobré přidat argument */compress*. Díky němu *cabwiz* instalační balík zkomprimuje. Při vytváření této distribuce došlo k velkým úsporám ve velikosti výsledného balíčku. Balíček bez komprese měl velikost 10,5 MB. Po kompresi už pouhých 2,4 MB.

Kapitola 5

Instalace a testy

Prvním krokem testování je zajisté instalace Perlu na zařízení. Po části instalace se podíváme na testování funkčnosti interpretru. Pro testovací účely bylo vytvořeno jednoduché testovací rozhraní, kterým si může uživatel ověřit, zda je Perl funkční. V následujících několika odstavcích se ho pokusím stručně představit a vysvětlit princip jeho fungování. Další sekce této kapitoly budou věnovány samotným testům a to jak v prostředí Microsoft Device Emulator, tak i na reálných zařízeních.

5.1 Instalace Perlu na zařízení

Všechny potřebné soubory jsou dostupné na přiloženém CD. Pro běh Perlu na zařízení potřebujeme pouze rozbalit archiv *perl512m.zip* na kartu telefonu a zkopírovat do něj soubor *perl-settings.CAB*. Ten posléze nainstalujeme jeho spuštěním na telefonu. Tím je instalace Perlu na zařízení dokončena.

Druhou možností jak Perl nainstalovat je zkopírování instalačního balíku *perl512m.CAB* na datovou kartu telefonu. Tuto instalaci stačí spustit a jako cílovou destinaci instalace vybrat datovou kartu. Tento postup však funguje pouze na reálných zařízeních. V případě instalace v emulátoru je potřeba použít první zmíněný postup.

Pokud chcete otestovat funkčnost Perlu, je dobré zkopírovat složku *Tests* na datovou kartu telefonu. Ta obsahuje jednoduché testovací rozhraní a několik základních testů. Toto rozhraní bude posáno dále v této kapitole, pro samotný běh interpretru však není nezbytné.

5.2 Testovací rozhraní

Základem rozhraní je skript *runTests.pl*, který se nachází v kořeni adresáře *Tests*. Tento skript předpokládá standardní umístění Perlu (`\Storage Card\Perl512m\bin\perl.exe`). Adresář *Tests* by měl být uložen přímo na paměťové kartě. Tato opatření jsou nutná kvůli tomu, že ve Windows Mobile nefungují relativní cesty. Pokud je však umístění Perlu nebo adresáře testů jiné, dá se snadno nastavit přímo ve skriptu. Na šestém řádku jde přenastavit absolutní cesta k Perlu pomocí změny proměnné *\$perlPath* a na sedmém je možné změnit umístění testů modifikací proměnné *\$testPath*.

Při testování stačí spustit skript *runTests.pl*, který spustí všechny testovací skripty. Výstupy skriptů se zapisují do výstupních souborů v adresáři *Results*. Jména výstupních souborů jsou stejná jako u skriptů, namísto přípony *.pl* však mají příponu *.out*. Ty jsou pak porovnány se soubory předpokládaného výstupu a uživatel je srozuměn s úspěchem

či neúspěchem jednotlivých testů na standardní výstup. Díky tomu, že adresář `Results` obsahuje výstupní a referenční soubory, může uživatel snadno dohledat, kde nastala chyba.

5.2.1 Definice vlastních testů

Základní sada testů se dá jednoduše rozšířit o další testy, kterými si uživatel může ověřovat funkčnost konstrukcí, které chce ve skriptech používat. Vytvoření vlastního testu a jeho přidání do testovací sady se skládá ze tří kroků.

Prvním krokem je vytvoření testovacího skriptu. Ten je potřeba nahrát do kořenového adresáře testů. Uživatel si může v testu nadefinovat libolné konstrukce a použití knihoven. Ve skriptu jsou potřeba výsledky vypisovat tak, aby mohly být po spuštění porovnávány. Pro výpis se však nepoužije standardní výstup, ale soubor, který je skriptu předán jako argument při spuštění. Příklad výpisu: `print OUT, $promenna`.

V druhém kroku je třeba vytvořit soubor s referenčním výstupem. Ten se musí jmenovat stejně jako skript, je však uložen v adresáři `Results`. Přípona tohoto souboru není `.pl`, ale `.expected`. Soubor musí obsahovat přesnou definici výstupu, který uživatel očekává. Na základě porovnání referenčního souboru se skutečným výstupem skriptu je uživatel seznámen s úspěchem nebo neúspěchem daného testu.

Posledním krokem je definování příslušného testu v souboru `test`, kde uživatel zapíše jméno testu který chce přidat. Díky tomuto mechanismu se dají testy snadno přidávat. V případě, že uživatel nechce nějaký skript spustit, stačí zakomentovat řádek napsáním znaku hash (`#`) na začátek řádku.

Na závěr uvedu ještě příklad obsahu jednoduchého skriptu, jeho referenčního výstupu a nastavení v souboru `test`.

Příklad testovacího skriptu `tryPrint.pl`:

```
use strict;
my $outFile = shift;
open(OUT,">",$outFile) or die $!;
print OUT, "My first test script."; close OUT;
```

Příklad referenčního souboru `tryPrint.expected`:

```
My first test script.
```

Příklad souboru `test`:

```
#Test files
someTest.pl
tryPrint.pl
#commentedTest.pl
```

5.3 Sada testů

Sada obsahuje několik testů, které ověřují funkčnost jazyka Perl na platformě Windows Mobile. Testy jsou zaměřeny především na základní funkčnost. Obsahují především práci s čísly a číselnými soustavami, řetězci, vestavěnými funkcemi a regulárními výrazy. Jejich rozsah lze snadno rozšířit pomocí vytvoření dodatečných testů, jak bylo popsáno v sekci [5.2](#).

5.4 Testy v Microsoft Device Emulator

V emulátoru byly testy spuštěny na třech verzích systému: Windows Mobile 5, Windows Mobile 6 a Windows Mobile 6.5. Ve všech verzích proběhla instalace nastavení v pořádku a ihned po instalaci a rozbalení archivu s Perlem bylo možné spouštět jednotlivé skripty. Nebylo možné otestovat instalaci celého Perlu. Ta vždy po chvíli selhala. Použit byl tedy pouze instalační balík nastavení a následné rozbalení archivu perl512m.zip do složky, která byla k emulátoru připojena jako datová karta.

Pokud není spuštěn žádný skript, ale přímo aplikace perl.exe, zobrazí se vyskakovací okno s hláškou. To je vidět na obrázku 5.1.



Obrázek 5.1: Perl spuštěný v Microsoft Device Emulator

5.5 Testy na hardwarových zařízeních

Stejně jako v emulátoru instalaci ani následné testování neprovázely žádné větší problémy. Narozdíl od emulátoru se na reálném hardware mohla testovat kompletní instalace Perlu balíkem perl512m.CAB. Testovány byly obě varianty popsané v 5.1. S výjimkou zařízení Acer n30, kde neproběhla správně instalace. Tyto problémy jsou blíže popsány v sekci 5.5.3

Pro testování byly použity tři zařízení HTC Touch Diamond 2, Mio A701 a Acer n30, blíže popsané níže. Každý z těchto strojů má jinou verzi operačního systému. Mio má verzi Windows Mobile 5.0, Acer používá Windows Mobile 2003 a na HTC byla nainstalovaná verze 6.5. Původní systém na HTC je Windows Mobile 6.1 Professional. Na testovacím kuse byla ovšem nainstalovaná novější verze distribuovaná jako oficiální update firmy HTC. Tím jsem mohl vyzkoušet, jestli Perl funguje ve více verzích operačního systému na reálném hardware.

5.5.1 Testy na HTC Touch Diamond 2

Na obrázku 5.2 vidíte HTC Touch Diamond 2. Obrázek a následná tabulka 5.1 je převzata ze stránek výrobce [1].



Obrázek 5.2: HTC Touch Diamond 2

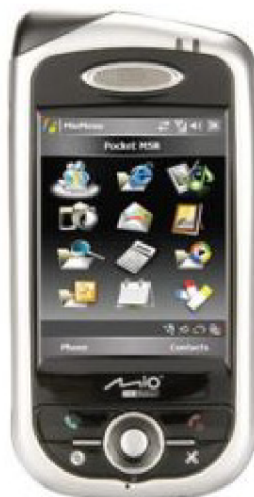
Procesor	Qualcomm MSM7200A, 528 MHz
Operační systém	Windows Mobile 6.5
Paměť	ROM: 512 MB RAM: 288 MB
Rozlišení	480 X 800 WVGA

Tabulka 5.1: Základní specifikace použitého HTC Touch Diamond 2

Na tomto zařízení proběhly všechny testy v pořádku. Po nainstalování Perlu bylo ihned možné spustit testy. Při namátkovém zkoušení dalších jednoduchých skriptů jsem neobjevil žádný závažný problém. Distribuce byla na Windows Mobile 6.5 plně funkční.

5.5.2 Testy na Mio A701

Na obrázku 5.3 vidíte Mio A701. Obrázek a následná tabulka 5.2 je převzata z článku o tomto zařízení na webu ZDnet.com [14].



Obrázek 5.3: Mio A701

Procesor	Intel PXA270, 520 MHz
Operační systém	Windows Mobile 5.0
Paměť	ROM: 128 MB RAM: 64 MB
Rozlišení	320 X 240

Tabulka 5.2: Základní specifikace použitého Mio A701

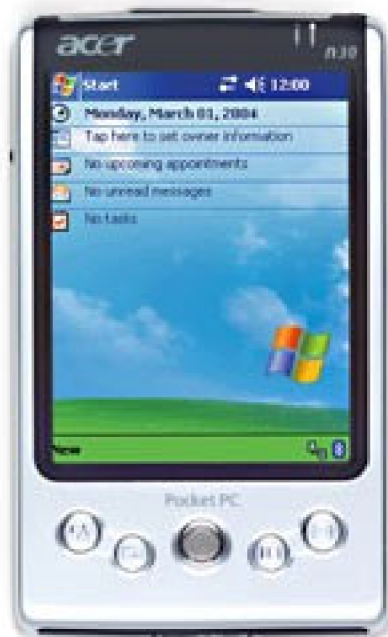
Na tomto zařízení proběhly všechny testy v pořádku. Po nainstalování Perlu bylo ihned možné spustit testy. Při namátkovém zkoušení dalších jednoduchých skriptů jsem neobjevil žádný závažný problém. Distribuce byla na Windows Mobile 5.0 plně funkční.

5.5.3 Testy na Acer n30

Na obrázku 5.4 vidíte Mio A701. Obrázek a tabulka 5.3 byly převzaty z článku o tomto zařízení na webu ZDnet.com [13].

Procesor	Samsung S3C2410, 266 MHz
Operační systém	Windows Mobile 2003
Paměť	ROM: 32 MB RAM: 64 MB

Tabulka 5.3: Základní specifikace použitého Acer n30



Obrázek 5.4: Acer n30

Na zařízení se vyskytlo několik problémů způsobených verzí systému a nastavením telefonu. Datová karta v telefonu nebyla pojmenovaná *Storage Card*, ale *SD-MMC Card*. To způsobilo nefunkčnost instalace nastavení. Pro opravu jsem použil program *Total Commander for PocketPC/WinCE*. Jde o volně dostupný software, který je k dispozici ke stažení na stránkách výrobce [5]. Obsahuje plugin pro editaci registrů. V sekci 4.6 je nastavení registrů popsáno blíže.

Dalším problémem byla nemožnost instalace distribuce. Na telefonu fungovala pouze instalace samotného nastavení, distribuce Perlu musela být rozbalena na datovou kartu z archivu perl512m.zip.

Po úpravách registrů už vše běželo bez problémů. Tím jsem ověřil, že distribuce je funkční i na verzi Windows Mobile 2003. Z tohoto testu ovšem vyplynulo, že se mohou instalovat pouze nastavení. Při pokusu o instalaci kompletní distribuce nastala chyba.

Kapitola 6

Závěr

Cílem této bakalářské práce bylo vytvořit funkční port interpretru jazyka Perl pro operační systémy Windows Mobile. V první řadě bylo zapotřebí seznámit se se zdrojovými kódy interpretru a zjistit jakými prostředky je možné tyto kódy přeložit pro cílovou platformu.

V první části byl popsán operační systém Windows Mobile, vlastnosti jazyka Perl a architektura ARM, která je v dnešní době používána pro výrobu procesorů mobilních zařízení. Další část práce postupně rozebírá použité nástroje pro překlad.

Na tuto část navazuje již samotný překlad interpretru rozdělený do několika dílčích kroků. Ty vysvětlují jak postupovat při jednotlivých částech překladu. V rámci této části vznikly úpravy v kódu interpretru a byly vytvořeny nové soubory Makefile jak pro samotný překlad Perlu pro Windows Mobile, tak i pro knihovny CeLib a W32console, které jsou pro překlad potřebné. Úpravy byly provedeny tak, aby bylo možné podle stejného postupu přeložit i budoucí verze interpreteru.

Poslední částí práce je instalace distribuce vzniklé aplikací úprav popsaných v předchozích kapitolách a testování dosaženého výsledku na virtuálních strojích i hardwarových zařízeních.

Literatura

- [1] HTC Touch Diamond 2 – Specification. [online]. 2011 [cit. 2011-04-25].
URL <http://www.htc.com/europe/product/touchdiamond2/specification.html>
- [2] Vytváříme instalační CAB soubory (nejen pro vývojáře). [online]. 2004-17-05, [cit. 2011-05-07].
URL <http://www.ce4you.cz/articles/detail.asp?a=97>
- [3] Windows Mobile. [online], 2011-01-30, [cit. 2011-01-30].
URL http://en.wikipedia.org/wiki/Windows_Mobile
- [4] The Perl Programming Language. [online], [cit. 2011-05-10].
URL <http://www.perl.org>
- [5] Total Commander – WinCE/PocketPC. [online], [cit. 2011-05-10].
URL <http://www.ghisler.com/ce.htm>
- [6] Perl for WinCE devices. [online], Poslední aktualizace 2006-11-06, [cit. 2011-05-10].
URL <http://perlce.sourceforge.net>
- [7] COZENS, S.: *Beginning Perl*. Wrox Press, 2000, ISBN 1861003145.
URL <http://www.perl.org/books/beginning-perl/>
- [8] FURBER, S.: *ARM system-on-chip-architecture*. Addison Wesley, druhé vydání, 2000, ISBN 9780201675191.
- [9] KEUCHEL, R.: [online], Poslední aktualizace 2003-03-23, [cit. 2011-04-10].
URL <http://sourceforge.net/projects/perlce/files/PerlCE%20support%20files/ce-lib-3.11/>
- [10] KEUCHEL, R.: [online], Poslední aktualizace 2003-06-10, [cit. 2011-04-10].
URL <http://sourceforge.net/projects/perlce/files/PerlCE%20support%20files/ce-lib-3.11/>
- [11] KRAZIT, T.: ARMed for the living room. [online]. 2006-04-03, [cit. 2011-05-07].
URL http://news.cnet.com/ARMed-for-the-living-room/2100-1006_3-6056729.html
- [12] ROBINSON, D.: Windows Phone 7 will not run Windows Mobile apps – V3.co.uk – formerly vnunet.com. 2011-01-30, [cit. 2011-01-30].
URL <http://www.v3.co.uk/v3/news/2259001/windows-phone-run-windows>
- [13] VOGEL, S.: Acer n30 Review. [online]. 2004-06-21, [cit. 2011-04-25].
URL <http://www.zdnet.co.uk/reviews/pdas/2004/06/21/acer-n30-39158269/>

- [14] VOGEL, S.: Mio A701 Review. [online]. 2006-03-31, [cit. 2011-04-25].
URL <http://www.zdnet.co.uk/reviews/smartphones/2006/03/31/mio-a701-39260600/>
- [15] WALL, L.; CHRISTIANSEN, T.; ORWANT, J.: *Programming Perl*. O'Reilly & Associates, Inc., třetí vydání, 2000, ISBN 0-596-00027-8.

Dodatek A

Obsah CD

- Zdrojový text práce v jazyku L^AT_EX
- Archiv zdrojových souborů Perlu verze 5.12.3
- Archiv zdrojových souborů a zkompileovaných dynamických knihoven knihovny CeLib verze 3.11
- Archiv zdrojových souborů a zkompileovaných objektových souborů knihovny W32console verze 1.3
- Instalační balík distribuce Perlu 5.12.3 pro Windows Mobile
- Instalační balík nastavení distribuce Perlu 5.12.3 pro Windows Mobile
- Skript pro vytváření informačních souborů.
- Archiv distribuce Perlu 5.12.3 pro Windows Mobile
- Složka s testy na mobilních zařízeních
- Složka se změněnými zdrojovými soubory Perlu, CeLib a W32Console