

Český průvodce programem Octave

Michal Just

Bakalářská práce
2006



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2005/2006

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal JUST**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Český průvodce programem Octave**

Zásady pro vypracování:

1. Provedte literární rešerši týkající se tvorby web stránek.
2. Seznamte se s prací v programu GNU Octave.
3. Vypracujte materiál v češtině obsahující charakteristiku tohoto programu a referenční příručku základních příkazů.
4. Naprogramujte ukázkové příklady demonstrující základní možnosti programu.
5. Vámi vytvořené studijní podklady převedte na formát HTML a umístěte na samostatné CD-ROM jako přílohu bakalářské práce.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

SELHOFER, Huber, OLIVER, Marcel. Introduction to GNU Octave, International University Bremen, Německo, 2003

(<http://math.iu-bremen.de/oliver/teaching/iub/resources/octave/octave-intro.pdf> [27.01.2006])

EATON, John W. Octave FAQ, 1998

(<http://www.mhatt.aps.anl.gov/dohn/software/octave/Octave-FAQ.pdf> [27.01.2006])

http://www.delorie.com/gnu/docs/octave/octave_toc.html [27.01.2006]

<http://octave.sourceforge.net/> [27.01.2006]

<http://www.kosek.cz/clanky/html/index.html> [27.01.2006]

Vedoucí bakalářské práce:

Ing. Karel Perůtka

Ústav řízení procesů

Datum zadání bakalářské práce:

14. února 2006

Termín odevzdání bakalářské práce:

16. června 2006

Ve Zlíně dne 14. února 2006



prof. Ing. Vladimír Vašek, CSc.
pověřený děkan



doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Tato bakalářská práce se zabývá realizací českého průvodce k programu Octave, který je volně dostupnou variantou komerčního softwaru Matlab. Hlavním cílem je vytvořit uživatelsky příjemné a informačně uspokojující WWW stránky, které obsahují překlad větší části originální dokumentace k programu. Druhou částí práce je vytvoření referenční příručky, která má stejně jako HTML stránka, (s menším komfortem interaktivity) pomoci případným zájemcům o používání tohoto programu v orientaci a také má poskytnout dostatečný základ pro efektivní práci s tímto druhem softwaru.

Klíčová slova: Octave, Matlab, internet, tvorba www, HTML, CSS, XHTML, XML, HTTP

ABSTRACT

This bachelor thesis deal with the realization of Czech guide for Octave programme which is freely accessible alternate for commercial Matlab software. The main aim was to create user-friendly WWW pages with satisfactory amount of information which contain almost all information from original documentation for the programme. Second part of this work was to create the referential handbook which as well as HTML page (with less confort of interactivity) have to help possible users interested in programming with this program with orientation and it should afford sufficient base for effective work with this kind of software.

Keywords: Octave, Matlab, Internet, production of www, HTML, CSS, XHTML, XML, HTTP

Rád bych na tomto místě poděkoval vedoucímu bakalářské práce Ing. Karlu Perůtkovi za jeho cenné rady, věcné podněty a celkově za odborné vedení mé bakalářské práce. Dále Ing. Tomáši Dulíkovi za odbornou výuku tvorby www stránek, která prohloubila mé znalosti týkající se této problematiky a také se stala odrazovým můstkem pro mou práci.

V neposlední řadě bych chtěl poděkovat rodinným členům za projevenou morální podporu, která dopomohla vzniku tohoto projektu.

Motto

“
Vzdělání není pouhým nashromážděním jednotlivých vědomostí,
jako není těstem mouka, voda, sůl, kvasnice atd. dohromady naházené.

”

TOMÁŠ GARRIGUE MASARYK (* 1850 – † 1937)

Souhlasím s tím, že s výsledky mé bakalářské práce může být naloženo podle uvážení vedoucího bakalářské práce a ředitele ústavu. V případě publikace budu uveden jako spoluautor.

Prohlašuji, že tuto bakalářskou práci jsem vypracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 15. 06. 2006

.....

Michal Just

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 INTERNET	10
1.1 HISTORIE	10
1.2 PRINCIP FUNGOVÁNÍ.....	11
1.3 MOŽNOSTI VYUŽITÍ	11
2 TECHNOLOGIE PRO TVORBU WWW STRÁNEK	13
2.1 HTML.....	13
2.1.1 Vývoj jazyka	13
2.1.2 Verze jazyka.....	13
2.1.3 Koncepce.....	14
2.1.4 Struktura dokumentu.....	15
2.1.5 Druhy značek	15
2.1.6 Příklad zdrojového kódu	16
2.1.7 Parsování v prohlížečích	17
2.1.8 Budoucnost HTML	18
2.2 KASKÁDOVÉ STYLY - CSS.....	19
2.2.1 Co to je?	19
2.2.2 Výhody.....	20
2.2.3 Práce se styly.....	20
2.3 JAVASKRIPT	24
2.3.1 Vznik	24
2.3.2 Syntaxe	24
2.3.3 Spouštění skriptů.....	25
2.4 XHTML.....	25
2.5 XML	26
2.5.1 Vlastnosti XML.....	26
2.5.2 Syntaxe XML	29
2.5.3 Aplikace XML	30
2.5.4 Verze XML	30
2.6 HTTP	31
2.6.1 Činnost protokolu.....	31
2.6.2 Ukázka komunikace	32
2.6.3 Dotazovací metody.....	33

II	PRAKTICKÁ ČÁST	34
3	SEZNÁMENÍ S PRACÍ V GNU OCTAVE	35
3.1	INSTALACE	35
3.2	SPOUŠTĚNÍ A UKONČENÍ	35
3.3	PRÁCE S PROGRAMEM.....	36
4	UKÁZKOVÉ PŘÍKLADY DEMONSTRUJÍCÍ ZÁKLADNÍ MOŽNOSTI PROGRAMU	37
4.1	VYTVOŘENÍ MATICE	37
4.2	VÝPIS PROMĚNNÉ	38
4.3	POČETNÍ OPERACE	38
4.4	VYKRESLOVÁNÍ GRAFŮ	39
4.4.1	2D grafy	39
4.4.2	3D grafy	40
4.5	NÁPOVĚDA	42
5	TVORBA HTML REFERENČNÍ PŘÍRUČKY	43
5.1	PŘÍPRAVA	43
5.1.1	Programové možnosti tvorby internetových stránek.....	43
5.1.2	Výběr softwaru	43
5.1.3	Srovnání efektivity vybraných nástrojů	44
5.2	VLASTNÍ PRÁCE NA HTML REFERENČNÍ PŘÍRUČCE.....	47
5.2.1	Umístění na internetu	48
5.2.2	Počátek práce	48
5.2.3	Grafické rozvržení.....	48
5.2.4	Koncepce stránek	49
5.2.5	Propojení hypertextovými odkazy, hlavní nabídka.....	51
5.2.6	Ukázkový soubor zdrojového kódu	52
5.2.7	Statistické údaje	56
	ZÁVĚR	57
	SEZNAM POUŽITÉ LITERATURY	58
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	59
	SEZNAM OBRÁZKŮ	60
	SEZNAM PŘÍLOH	61

ÚVOD

V 50. letech kdy se začaly budovat základy Internetu, nikdo ze zakladatelů netušil, co doopravdy tvoří. Jednalo se jen o pokus propojit jistý počet univerzit ve Spojených státech – projekt podporovaný armádou. Původně se předpokládalo, že tímto všechno skončí, avšak nestalo se tak. Síť rostla dál a na její základ se začalo připojovat nezměrné množství dalších sítí až došlo k propojování v rámci jednotlivých států a později kontinentů. Zanedlouho propojila celý svět do jedné velké pavučiny.

Jedním z impulsů k rozvoji Internetu byla služba WWW (World Wide Web). Zkratka znamená v překladu přibližně něco jako "síť obepínající planetu" a přesně vystihuje rozšíření této sítě. Tato služba vznikla jako řešení umožňující sdílet informace textového charakteru, a vyvinuté pro potřeby relativně uzavřené komunity fyziků. Byla ovšem tak vypracována, že ji začali používat všichni uživatelé Internetu.

Pokud však chceme přispět vlastní internetovou stránkou, musíme proniknout do tajů zkratk HTML a HTTP. Nejedná se však o vůbec nic složitého. Jazyk HTML (HyperText Markup Language) je charakterizován množinou značek a jejich atributů pro danou verzi definovaných. Mezi tyto značky se uzavírají části textu dokumentu a tím se určuje význam (sémantika) obsaženého textu. A HTTP (HyperText Transfer Protocol) je protokol používaný pro přenos těchto dokumentů.

Cílem mé bakalářské práce ve srozumitelné formě vypracovat studijní podklady k programu Octave, který je volně dostupnou variantou komerčního softwaru Matlab. Dle zadání práce tyto materiály budou vyhotoveny ve dvou podobách. Jedna koncipovaná jako referenční příručka a druhá jako HTML internetová stránka, která nabízí více interaktivity a také možnost uživatelsky příjemnějšího a graficky estetičtějšího řešení.

I. TEORETICKÁ ČÁST

1 INTERNET

Internet je jistě slovo dnešní doby. Je skloňováno ve všech pádech. Už i „nepočítačová“ veřejnost si začíná uvědomovat význam tohoto nového informačního a komunikačního média. Na rozdíl od rozhlasu a televize může totiž každý pomocí Internetu oslovit doslova celou planetu. Vy můžete komunikovat s celým světem a celý svět může komunikovat s vámi. Geografické hranice ztrácejí smysl. Práce s Internetem bude jistě již zanedlouho patřit k základním dovednostem člověka 21. století.

1.1 Historie

Předchůdcem a také základem Internetu se stala síť Ministerstva obrany USA ARPANET. Tato síť vznikla v roce 1969 a podílela se na ní organizace ARPA (Advanced Research Project Agency). Tímto si USA chtělo zajistit prvenství ve vědě a technologii aplikované ve vojenské praxi.

Hlavní myšlenkou při zakládání tohoto úřadu byla decentralizace počítačové sítě, která nebude mít jediný „kritický bod“. Jinými slovy tato síť byla navržena tak, aby byla schopna fungovat i v případě výpadku jejích jednotlivých částí, a aby neměla žádnou hlavní část, která by při zničení znemožňovala funkci zbytku sítě. Po vzniku této sítě se ARPA přejmenovala na DARPA (Defence Advanced Research Project Agency). Síť spojovala 4 počítače, přičemž hlavní uzel byl na University of California v Los Angeles, který byl vlastně prvním uzlem v historii Internetu a vznikl 6. v září 1969. Druhý uzel vznikl o měsíc později v Stanford Research Institute (SRI). Později byly vytvořeny další uzly v UC v Santa Barbaře a v University of Utah. Od roku 1969 se datuje vznik Internetu, lépe řečeno, jeho principu a technologie. Tedy samotný název Internet se samozřejmě ještě nepoužíval. Postupně během dalších let vznikaly různé sítě se specifickým zaměřením, avšak pracující na stejném principu, tedy vzájemně slučitelné.

Jednou z prvních funkcí sítě byla možnost posílat a přijímat elektronickou poštu a soubory. Další sítě vznikaly především ve vyspělých zemích kde byly dostatečné finance. Rychlost přenosu dat rostla. V roce 1986 byla založena páteří síť dosahující rychlosti přenosu až 56 kb/s, zahrnující 5 počítačových středisek. Tato střediska podnítila doslova explozi nově připojených počítačů na Internet. Připojovaly se další země, mezi nimi v roce 1991 i Česká Republika.

Roku 1992 došlo k založení společnosti zabývající se historií, vývojem, trendy a etikou Internetu. Počet hostitelských počítačů překročil jeden milion. V roce 1993 se zakládá instituce Internet Society (ISOC), což je dobrovolná organizace. ISOC jmenuje skupinu odborníků, která se nazývá Rada pro architekturu sítě Internet . Tato rada se pravidelně schází, aby schválila standardy a plán přidělování zdrojů, jako jsou například adresy počítačů, provádí registraci doménových jmen apod.

Na Internetu již můžeme sledovat audio i video vysílání prvních rozhlasových a televizních stanic. Připojují se obchodní společnosti a média, která do té doby neměla s počítači nic společného a nastává informační exploze. V roce 1994 slaví Internet 25. výročí. Roku 1995 se služba WWW dostává na první místo v počtu přenesených dat. Objevuje se první oficiální špionáž, namířená proti nelegálním výrobcům mobilních telefonů a dalšího elektronického zařízení, vedená americkou tajnou službou prostřednictvím Internetu. Rok 1996 přináší První veletrh Internetové technologie.

V současné době se Internet u nás stává stále více součástí běžného života. A to hlavně prostřednictvím elektronické pošty. Dnes je již zcela běžnou záležitostí uvádět na své vizitce též adresu elektronické pošty, vlastní stránky (homepage) nebo společností, které přístup k Internetu nabízejí, dochází i k poklesu cen a vzniká tedy prostředí s konkurencí a začíná boj o zákazníky. Budoucnost se zaměřuje ne více interaktivní práci uživatele s Internetem.

1.2 Princip fungování

Jak to vlastně všechno funguje? Každý počítač má svou specifickou adresu. Žádný jiný počítač nemůže mít stejnou adresu. Veškerá data se do PC dostávají prostřednictvím serverů, velmi výkonných počítačů, které obsahují, mimo jiné, databázi adres počítačů a data na ně rozesílají. Z toho vyplývá, že k Internetu je připojeno více druhů počítačů. Žádný z těchto PC nemusí být neustále zapnutý ani trvale připojený k Internetu. Jinak je to ovšem se servery, které tvoří páteř celého Internetu. Ty musí být v chodu neustále a trvale připojeny.

1.3 Možnosti využití

Co vlastně Internet nabízí? Především obrovské, nepřehledné množství informací. Ať už toužíte po adrese na mapě, odjezdu vlaků, prodejně elektroniky, životopisu vašeho

oblíbeného zpěváka nebo tisíce dalších věcí, najde vždy cokoli si vymyslíte. Připojením ke službě „chat“ si můžete živě povídat s mnoha jinými lidmi z celého světa. Další možností Internetu je e-mail (elektronická pošta). Oproti klasické poště má řadu výhod. Je rychlá.

Na druhý konec světa putuje zpráva necelou minutu. Můžete do ní vložit obrázek, hudbu, krátký filmeček, atd. Problémem však je její utajení. Prochází několika jinými počítači, kde by si mohl nudící se slídlil přečíst všechno co jste napsali.

Celá kniha by nestačila k popsání toho, co Internet poskytuje. Člověk nemusí být počítačový expert, aby s ním mohl pracovat, vzdělávat se nebo se jen bavit. Takže stačí si jen sednout, zapnout PC, klik, klik a udělat první krok do obrovského, fascinujícího světa moderní komunikace.

V neposlední řadě na Internetu narazíte na www neboli „web“, což znamená pavučina. Jsou to vlastně stránky obsahující text a obrázky. Části textu mohou být zvýrazněné tzv. hypertext. Poklepáním myši se zobrazí další stránka. Je to jednoduché, pěkné a hlavně funkční. V textu najdete slovo barevně a graficky zvýrazněno (např. modré a podtržené). Pokud na toto slovo najedete ukazatelem myši, změní se ukazatel na malou ručku. Tomuto slovu se říká odkaz. Dal by se přirovnat ke skrytému ukazateli na jiné místo. Jestliže na toto slovo kliknete, objeví se stránka na kterou odkaz ukazoval. Na tomto principu funguje celá síť. Takto jsou pospojovány související informace a vytvářejí tak opravdu obrovskou informační síť, rozprostírající se po celé planetě... [1],[4]

2 TECHNOLOGIE PRO TVORBU WWW STRÁNEK

2.1 HTML

HTML je zkratka z anglického HyperText Markup Language, značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci stránek na Internetu.

Jazyk je podmnožinou dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

2.1.1 Vývoj jazyka

V roce 1989 spolupracovali Tim Berners-Lee a Robert Caillau na propojeném informačním systému pro CERN, výzkumné centrum fyziky poblíž Ženevy ve Švýcarsku. V té době se pro tvorbu dokumentů obvykle používal TeX, Postscript a také SGML. Berners-Lee si uvědomoval, že potřebují něco jednoduššího a v roce 1990 byl tedy navržen jazyk HTML a protokol pro jeho přenos v síti - HTTP (HyperText Transfer Protocol - přenosový protokol hypertextu).

V roce 1991 CERN zprovoznil svůj web. Současně NCSA (National Center for Supercomputer Applications) vybídlo Marca Andreessena a Erica Binu k vyvinutí prohlížeče Mosaic. Byl vyvinut v roce 1993 pro počítače PC a Macintosh a měl obrovský úspěch. Byl to první prohlížeč s grafickým uživatelským rozhraním. Došlo k velkému rozvoji webu a bylo nutné pro HTML definovat standardy.

2.1.2 Verze jazyka

Verze 2.0 zachycuje stav jazyka v polovině roku 1994. Standard vydala komunita IETF (Internet Engineering Task Force). Je to první verze, která odpovídá syntaxi SGML. Přidává k původní specifikaci interaktivní formuláře.

Verze 3.2 byla vydána v květnu 1996 a zachycuje stav jazyka v tomto roce. Připravovaná verze HTML 3.0 nebyla nikdy přijata jako standard, protože byla příliš složitá a žádná firma nebyla schopna naprogramovat jeho podporu ve svém prohlížeči. Standard už vydalo

konsorcium W3C, stejně jako následující verze. Přidává k jazyku tabulky, zarovnávání textu a stylové elementy pro ovlivňování vzhledu.

Verze 4.0 byla vydána 18. prosince 1997. Do specifikace jazyka přidala nové prvky pro tvorbu tabulek, formulářů a nově byly definovány rámy (frames). Tato verze se snaží dosáhnout původního účelu - prvky by měly vyznačovat význam (sémantiku) jednotlivých částí dokumentu, vzhled má být ovlivňován připojovanými styly. Některé prezentační elementy byly zavrženy.

Verze 4.01 opravuje některé chyby verze předchozí a přidává některé nové tagy. Je to poslední verze HTML, které se již dále nevyvíjí a je nahrazeno novějším XHTML, jehož základem je právě tato poslední verze HTML.

2.1.3 Koncepce

Jazyk HTML je od verze 2.0 aplikací SGML. Je charakterizován množinou značek a jejich atributů pro danou verzi definovaných. Mezi značky se uzavírají části textu dokumentu a tím se určuje význam (sémantika) obsaženého textu. Názvy jednotlivých značek se uzavírají mezi úhlové závorky ("*<*" a "*>*"). Část dokumentu uzavřená mezi značkami tvoří tzv. element (prvek) dokumentu. Součástí obsahu elementu mohou být další vnořené elementy. Atributy jsou doplňující informace, které upřesňují vlastnosti elementu.

Značky (také nazývané tagy) jsou obvykle párové. Rozlišujeme počáteční a koncové značky. Koncová značka má před názvem značky znak lomítka. Příklad pro označení odstavce:

```
<p>Text odstavce</p>
```

Některé značky jsou nepárové - neobsahují žádný text. Příklad pro vykreslení vodorovné čáry, s atributem pro tloušťku čáry 4 pixely:

```
<hr size=4>
```

Pro každou verzi existuje definice pravidel DTD (Document Type Definition). Od verze 4.01 musí být odkaz na deklaraci DTD v dokumentu uveden pomocí klíčového slova DOCTYPE. DTD definuje pro určitou verzi, které elementy je možné používat a s jakými atributy.

Dokument může mimo značkování obsahovat další prvky:

- Direktivy - začínají znaky „<!\“, jsou určeny pro zpracovatele dokumentu (prohlížeč)
- Komentáře - pomocné texty pro programátora, nejsou součástí obsahu dokumentu a nezobrazují se (prohlížeč je ignoruje). Příklad komentáře je níže.
- Kód skriptovacích jazyků
- Definice událostí a kód pro jejich obsluhu

2.1.4 Struktura dokumentu

Dokument v jazyku HTML má předepsanou strukturu:

- Deklarace DTD - je povinná až ve verzi 4.01, začíná direktivou <!DOCTYPE.
- Kořenový element - element html (značky <html> a </html>) reprezentuje celý dokument. Je nepovinný, ale je doporučeno ho používat.
- Hlavička elementu - jsou to metadata, která se vztahují k celému dokumentu. Definují např. název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl zobrazení. Hlavička je uzavřena mezi značky <head> a </head>.
- Tělo dokumentu - obsahuje vlastní text dokumentu. Vymezuje se značkami <body> a </body>.

2.1.5 Druhy značek

Značky můžeme z hlediska významu rozdělit na tři základní skupiny:

- Strukturální značky jsou ty, které rozvrhují strukturu dokumentu. Příkladem jsou odstavce (<p>), nadpisy (<h1>, <h2>). Dodávají dokumentu formu.
- Popisné (sémantické) značky popisují povahu obsahu elementu. Příklad nadpis (<title>) nebo adresa (<address>). Současný trend je orientován právě na sémantické značky, které usnadňují automatizované zpracovávání dokumentů a vyhledávání informací v záplavě dokumentů na webu. Vyvrcholením této snahy je v současné době jazyk XML.

- Stylistické značky určují vzhled elementu při zobrazení. Typickým příkladem je značka pro tučné písmo (``). Tento druh značek se nedoporučuje používat, trendem je používání kaskádových stylů oddělených od obsahu dokumentu. Problém je především v tom, že tyto značky jsou orientovány na obrazovku monitoru, problémy mohou nastat při používání dokumentu jiným způsobem - alternativní prohlížeče pro postižené (čtečky pro slepce), v mobilních zařízeních a podobně. Kaskádové styly umožňují definovat rozdílné zobrazení pro různá zařízení.

2.1.6 Příklad zdrojového kódu

Příklad HTML pro verzi 4.01:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<!-- toto je komentář -->
  <head>
    <title>Titulek stránky</title>
  </head>
<!-- tělo dokumentu -->
  <body>
    <H1>Nadpis stránky</H1>
    <p>Toto je tělo dokumentu</p>
  </body>
</html>
```

Pak při své reprezentaci v prohlížeči vypadá následovně:



Obr. 1 Reprezentace zdrojového kódu

2.1.7 Parsování v prohlížečích

Webové prohlížeče jsou programy, jejichž účelem je prezentovat dokument na zobrazovacím zařízení - převážně monitoru počítače. Dokument je prohlížečem načítán a rozkládán (parsován, syntaktická analýza) na jednotlivé elementy. Prohlížeč obsahuje tabulku značek, které podporuje. Tuto tabulku je možné omezit typem dokumentu (DTD), je-li deklarován.

Každému elementu je poté přiřazen styl (způsob zobrazení). Styly mohou být uvedeny ve stylovém předpisu. Vlastnosti stylů, které nejsou předepsány, doplní prohlížeč podle implicitního stylu, který má zabudován. Některé prohlížeče umožňují uživateli implicitní styly definovat.

Novější prohlížeče pracují obecně ve dvou základních režimech:

- Standardní režim - je dodržována specifikace verze HTML, deklarovaná v dokumentu. Elementy v dokumentu musí odpovídat verzi, neznáme elementy jsou považovány za chybu a nezobrazují se. Netolerují se syntaktické chyby.
- Quirk mód - nestandardní režim. Prohlížeč se snaží „napravit“ chyby v syntaxi dokumentu, domýšlí si chybějící koncové párové značky, neznámé elementy zobrazuje implicitním formátem.
- Režim se stanoví obecně podle typu dokumentu, ale rozhodování je dost složité (označuje se výrazem doctype sniffing). Účelem je zachování zpětné kompatibility pro dokumenty, které nedodržují definici HTML. Tato vlastnost ale umožňuje vytvářet nestandardní dokumenty - autory zatím nic nenutí pracovat bez chyb podle specifikace.

Prohlížeče mají různé implicitní styly zobrazování a chovají se různě v quirk módu. Stejný dokument se tedy může v různých prohlížečích zobrazovat různě, rozdíly jsou i mezi verzemi prohlížeče stejného výrobce. Současné chování prohlížečů je výsledek vývoje, kdy si výrobci jednotlivých prohlížečů přizpůsobovali definici HTML podle svých potřeb a prohlížeče podporovaly nestandardní elementy a syntaxi. Řada těchto „vylepšení“ byla následně přejímána do standardů, a některé zase v dalších verzích vyřazeny.

2.1.8 Budoucnost HTML

Vývoj jazyka HTML byl ukončen verzí 4.01. Dalším pokračováním vývoje jazyků pro psaní dokumentů na webu je jazyk XHTML.

Důvodů pro tuto změnu je celá řada. Překotný vývoj prohlížečů umožnil vznik obrovského množství dokumentů, které neodpovídají specifikaci HTML. Současné prohlížeče tolerují chybné značkování (neuzavřené, překřížené, či špatně vnořené elementy) a řadu dalších chyb a při parsování se s těmito chybami snaží vyrovnat („myslí“ za uživatele), zpravidla však ale ne dle představ tvůrce. Tento princip je ale příliš náročný pro nový druh aplikací, které vznikají pro malá méně výkonná zařízení - mobilní telefony, přenosné počítače (palmtopy) atd. Tyto aplikace potřebují přesně definovanou strukturu s přesnými pravidly. To poskytuje jazyk XML, ale HTML není aplikací XML a těmito novým požadavkům

nevyhovuje. Pokud by však nové verze prohlížečů začaly vyžadovat přesně napsané HTML dokumenty bez chyb, pravděpodobně většina dokumentů na webu by nešla zobrazit.

Dá se říct, že momentálně probíhá určitý spor mezi webovými vývojáři o to, zda přejít na XHTML, nebo dále používat HTML. Jsou různé argumenty pro a proti, souvisí také s používáním kaskádových stylů a se způsobem rozvržení stránky (tabulky versus pozicování styly) Tento argument je zavádějící, protože XHTML 1.0 a HTML 4.0 jsou si prakticky ekvivalentní, měnila se pouze syntaxe. Velký vliv při tomto rozhodování má kvalita nejrozšířenějších prohlížečů, stránky se musí navrhovat tak, aby šly zobrazit běžnými prostředky.

Vývoj tedy pokračuje jiným jazykem, ale specifikace HTML platí dál - je tedy možné vyvíjet nové aplikace a vytvářet dokumenty podle HTML 4.01 všude tam, kde tento jazyk stačí.

I HTML kód je možné psát tak, aby do značné míry vyhovoval novým požadavkům na dokumenty webu. [1],[4]

2.2 Kaskádové styly - CSS

Dnes v HTML existuje několik atributů a elementů, které ovlivňují pouze grafický vzhled. Použitím těchto atributů a elementů sice můžeme získat graficky atraktivní stránku, má to však řadu nevýhod.

Text stránky je mnohdy špatně strukturován, protože jednotlivé elementy jsou využívány účelově k dosažení určitých grafických efektů. Druhou velkou nevýhodou je velká pracnost - většina z vizuálních atributů musí být nastavována opakovaně u všech elementů.

2.2.1 Co to je?

Obě tyto nevýhody odstraňují kaskádové styly. CSS je zkratka pro anglický název Cascading Style Sheets, česky tabulky kaskádových stylů. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML.

Jazyk byl navržen standardizační organizací W3C. Byly vydány zatím dvě verze specifikace CSS1 a CSS2, pracuje se na verzi CSS3 (rok 2006).

Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí.

Styly umožňují definovat způsob zobrazení (druh a velikost písma, barvu, zarovnání apod.) každého elementu na stránce. Styl však není přímo součástí textu stránky a tak může být zápis stránky přehlednější a dobře strukturovaný. Navíc styly umožňují definovat jednotný vzhled určitého elementu v celém dokumentu jedním zápisem - nemusíme jej opakovat u každého elementu.

2.2.2 Výhody

Používání kaskádových stylů ve srovnání se samotným HTML v praxi přináší výhody:

- širší formátovací možnosti
- snadná tvorba a údržba konzistentního stylu
- oddělení struktury a stylu
- dynamická práce se styly
- formátování XML dokumentů
- větší kompatibilita alternativních webových prohlížečů
- kratší doba načítání stránky

2.2.3 Práce se styly

Stylový předpis se skládá z posloupnosti pravidel. Každé pravidlo určuje vzhled některého elementu dokumentu, nebo skupiny elementů. Pravidlo začíná tzv. selektorem, který specifikuje („adresuje“) skupinu elementů. Selektor je následován seznamem deklarácí, které určují vzhled vybrané skupiny elementů. Celý seznam je uzavřen ve složených závorkách a jednotlivé deklarace jsou odděleny středníkem (tj. za poslední deklarací středník už být nemusí).

Příklady použití

Práce se styly je velmi jednoduchá. Ukážeme si ji na jednoduchém stylu, který zajistí, aby všechny nadpisy první až třetí úrovně byly zobrazeny bezpatkovým písmem a modře.

Nadpisy první úrovně budou navíc ještě vycentrovány:

```
H1, H2, H3 { font-family: Arial, sans-serif; color: blue }
H1         { text-align: center }
```

Tento jednoduchý styl se skládá ze dvou pravidel. Každé pravidlo má dvě části - selektor (např. H1) a deklaraci (text-align: center). Všechny deklarace se skládají ze dvou částí:

- z vlastnosti (např. text-align)
- z její hodnoty (center)

Pravidla i deklarace lze navzájem různě sdružovat.

Další příklad stylu:

```
h1 {                               /* vzhled nadpisu první úrovně */
    margin: 5px;                   /* okraj šířky pixelů          */
    font-size: 12pt                /* velikost fontu 12 bodů    */
}
p  {                               /* styl odstavce           */
    text-align: center;           /* text centrovat          */
    line-height: 10pt;          /* výška řádku 10 bodů    */
}
```

S pomocí kaskádových stylů lze na stránce ovlivňovat nepřeberné množství věcí. Z těch důležitějších jmenujme například rozšíření stylů o řízení pozice elementů. Rozšíření přináší nové vlastnosti, které umožňují přesně řídit vzájemné umístění jednotlivých elementů na stránce. Mezi nejdůležitější nové vlastnosti patří left a top, které určují posunutí elementu. Jejich konkrétní interpretace však závisí na hodnotě vlastnosti position. Ta může nabývat dvou hodnot - absolute a relative, které rozlišují mezi absolutním a relativním určením pozice.

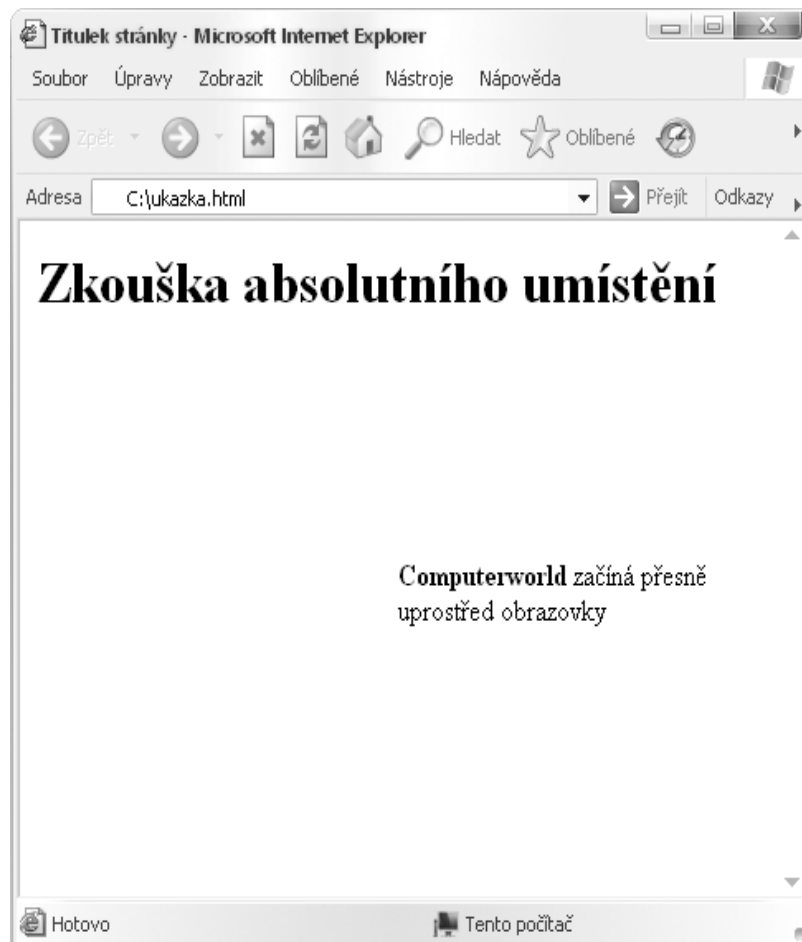
Absolutní určení pozice

Pokud použijeme absolutní určení pozice elementu, je tento element zobrazen zcela nezávisle na svém původním umístění. Jeho levý horní roh se posune do místa zadaného pomocí left a top. Souřadnice se vztahují k souřadnému systému nadřazeného elementu a tím je nejčastěji celý dokument. Jeho levý horní roh má souřadnice (0,0).

Jako hodnotu vlastností left a top můžeme použít buď pixely (px) nebo procento, které se vztahuje k šířce resp. výšce rodičovského elementu:

```
<BODY>
<H1>Zkouška absolutního umístění</H1>
<SPAN STYLE="position: absolute; left: 50%; top: 50%">
<B>Computerworld</B> začíná přesně uprostřed obrazovky</SPAN>
</BODY>
```

Pak při své reprezentaci v prohlížeči vypadá následovně:



Obr. 2 Reprezentace absolutního umístění

Relativní určení pozice

Pokud použijeme relativní určení pozice, element se pouze posune ze své přirozené pozice o délku určenou pomocí left a top. Jeho formátování přitom zůstane zachováno. Text v nadřazeném elementu zůstane zformátován tak, jako by posunutý element nebyl vůbec posunut. Malá ukázka:

Nějaký text

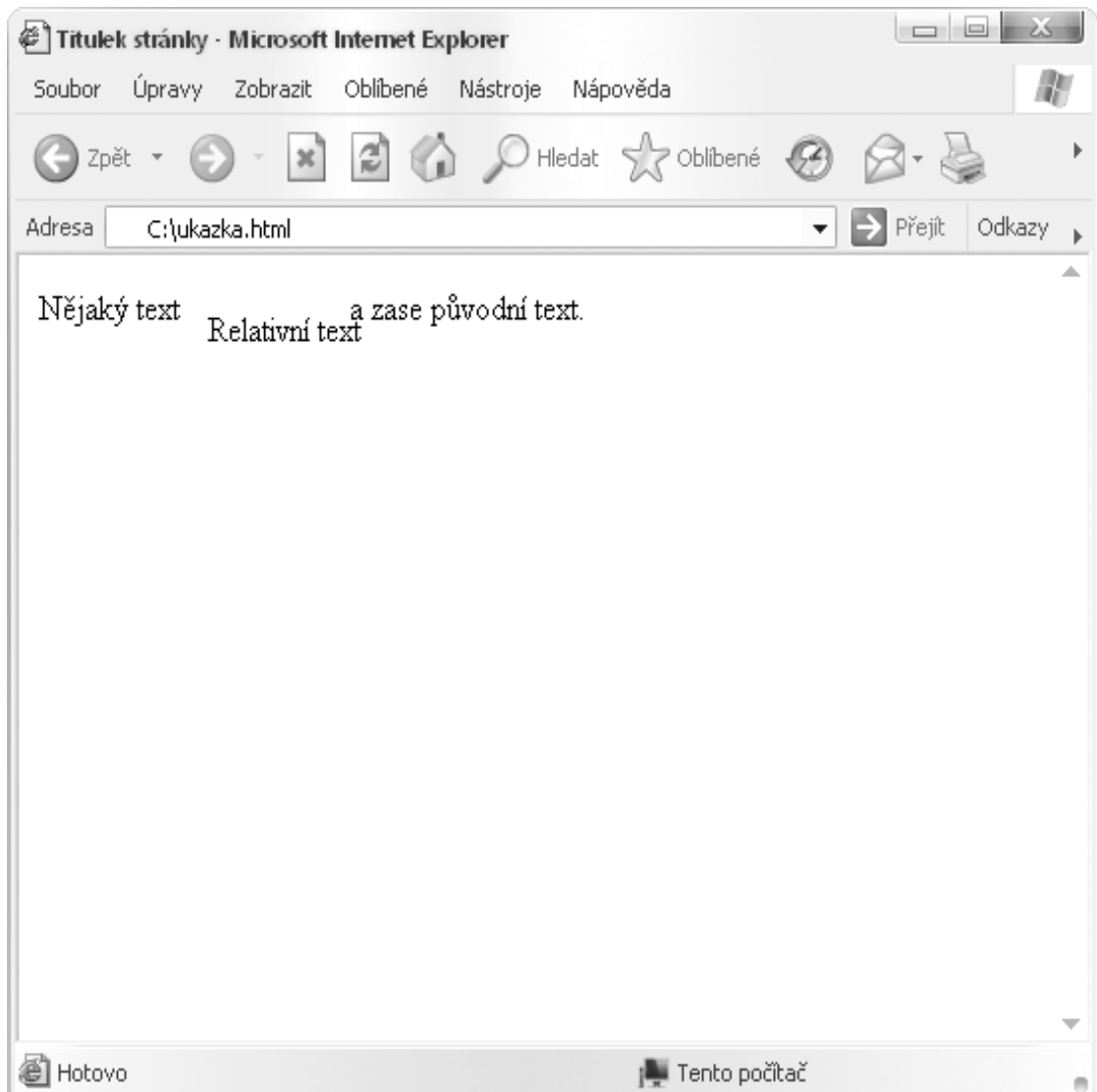
```
<SPAN STYLE="position: relative; left: 10px; top: 10px">
```

```
Relativní text</SPAN>
```

a zase původní text.

[1],[2]

Při své reprezentaci v prohlížeči pak vypadá následovně:



Obr. 3 Reprezentace relativního umístění

2.3 JAVASkript

JavaScript je jednoduchý multiplatformní, objektově orientovaný skriptovací programovací jazyk, který je možno zapisovat přímo do HTML stránky. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky. Pochází z vývojářské dílny firmy Netscape, ale dnes ho podporuje celá řada prohlížečů – ať už Internet Explorer, Mozilla či Opera. JavaScript byl původně obchodní název implementace společnosti Netscape, kde byl vyvíjen nejprve pod názvem LiveScript, ohlášen byl společně se společností Sun v prosinci 1995 jako doplněk k jazykům HTML a Java. Pro verzi firmy Microsoft je použit název JScript.

2.3.1 Vznik

Příčinou vzniku JavaScriptu byl požadavek na zvýšení uživatelského komfortu pro uživatele stránek. JavaScript můžeme použít například při vstupní kontrole dat vkládaných do

formulářů ještě předtím, než jsou vyplněné údaje odeslány na server. Kontrolu údajů nemusí provádět server a výsledkem je rychlejší odezva pro uživatele. Již ve svých prvních verzích umožnil i vytváření interaktivnějšího uživatelského rozhraní stránek. Stránka mohla reagovat na různé události vyvolané uživatelem a například doplňovat pole do formulářů nebo vyvolat otevření nového okna prohlížeče. Jsou jím také obvykle ovládané různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků. Tyto možnosti pak dále rozšířil nový koncept dynamického HTML.

2.3.2 Syntaxe

Jeho syntaxe patří do rodiny jazyků C/C++/Java. Slovo Java je však součástí jeho názvu pouze s marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript byl v červenci 1997 standardizován asociací ECMA (European Computer Manufacturers Association) a v srpnu 1998 ISO (International Standards Organization). Standardizovaná verze JavaScriptu je pojmenována jako ECMAScript a z ní byly odvozeny i další implementace, jako je například ActionScript.

Každý program se skládá z příkazů, které se provádí sekvenčně v pořadí, v jakém jsou zapísány. Jednotlivé příkazy se od sebe oddělují středníkem a je dobrým zvykem psát každý příkaz na samostatnou řádku - program je pak mnohem přehlednější.

2.3.3 Spouštění skriptů

Program v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu (tzv. na straně klienta), narozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele. Původně byl ale i JavaScript používán i na straně serveru. [1]

2.4 XHTML

Výše zmíněná specifikace HTML 4.0 byla později revidována, byly opraveny některé chyby, a stala se z ní specifikace HTML 4.01. Tato verze obsahuje kompletní popis jazyka a všechny následující specifikace jazyků pro tvorbu webu, které dosud vyšly, se na ni odkazují.

V další fázi bylo potřeba definovat HTML jako podmnožinu jazyka XML (eXtensible Markup Language - rozšiřitelný značkovací jazyk), který vychází ze SGML, avšak je „ořezán“ o některé zbytečné vlastnosti a některé nové vlastnosti naopak přidává. W3C prosazuje XML jako hlavní a jediný značkovací jazyk nejen pro web, proto byla reformulace HTML do XML logickým vyústěním této snahy. Nově vytvořený jazyk dostal jméno XHTML (eXtensible HyperText Markup Language - rozšiřitelný hypertextový značkovací jazyk) a jeho první specifikace se označuje XHTML 1.0. Tato specifikace je stejná jako specifikace HTML 4.01, jsou zde pouze integrována pravidla.

W3C chce XHTML prosazovat jako jediný jazyk pro definici webových stránek, bez ohledu na typ zařízení, které je zpracovává. Expanze webu na různá alternativní zařízení ukázala, že tato zařízení kvůli svým omezeným možnostem nemohou podporovat všechny vlastnosti XHTML, že podporují jen nějakou jeho část (podmnožinu). Tyto podmnožiny bylo třeba definovat a standardizovat, proto vznikla specifikace Modularization of XHTML.

Tato specifikace rozděluje všechny prvky XHTML 1.0 do modulů, ze kterých se následně skládají značkovací jazyky. Skupiny zabývající se webem v alternativních zařízeních mohou definovat nové moduly s prvky specifickými pro dané zařízení, mohou stávající moduly

modifikovat, ale hlavně mohou z modulů skládat nové kompletní značkovací jazyky, které vyhovují potřebám a možnostem interpretace webu na těchto zařízeních. Z těchto jazyků zatím W3C uznalo XHTML 1.1 (až na několik detailů se shoduje s XHTML 1.0 Strict, hlavní rozdíl spočívá v tom, že XHTML 1.1 je definováno pomocí modulů) a XHTML Basic (skládá se pouze ze základních modulů, k použití hlavně na mobilních telefonech, PDA a podobně). [1],[3]

2.5 XML

XML (eXtensible Markup Language, česky rozšiřitelný značkovací jazyk) je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Umožňuje snadné vytváření konkrétních značkovací jazyků pro různé účely a široké spektrum různých typů dat.

Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Jazyk umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, nezabývá se sám o sobě vzhledem dokumentu nebo jeho částí. Prezentace dokumentu (vzhled) se potom definuje připojeným stylem. Další možností je pomocí různých stylů provést transformaci do jiného typu dokumentu, nebo do jiné struktury XML.

Původní jazyk pro publikování HTML již přestal vyhovovat především pro svou složitost, která vznikla jeho postupným (a svévolným) rozšiřováním. Jazyk XML nemá žádné předdefinované značky (tagy, názvy jednotlivých elementů) a také jeho syntaxe je podstatně přísnější, než HTML.

2.5.1 Vlastnosti XML

Standardní formát pro výměnu informací

V dnešní době již není možné zasílat dokumenty v nějakém tvaru, který vyžaduje speciální software některé firmy, jako je např. formát DOC, XLS nebo T602. Je používána celá řada operačních a informačních systémů a není záruka, že každý uživatel vlastní příslušný software.

Je tedy potřeba používat nějaký jednoduchý otevřený formát, který není úzce svázan s nějakou platformou nebo proprietární technologií. Tím může být právě XML, který je

založen na jednoduchém textu a je zpracovatelný (v případě potřeby) libovolným textovým editorem.

Specifikace XML konsorcia W3C je zdarma přístupná všem. Každý tak může bez problémů do svých aplikací implementovat podporu XML. To je velký rozdíl oproti firemním formátům, k nimž není k dispozici žádná dokumentace a navíc se jedná v porovnání s XML o velice složité, často binární, formáty.

Mezinárodní podpora

XML hned od samého počátku myslel na potřeby i jiných jazyků než je angličtina. Jako znaková sada se implicitně používá ISO 10646 (také Unicode). V XML proto můžeme vytvářet dokumenty, které obsahují texty v mnoha jazycích najednou – můžeme přepínat mezi různými jazyky v jednom dokumentu. Současně je přípustné i jiné libovolné kódování (např. windows-1250, iso-8859-2), musí však být v každém dokumentu přesně určeno. Odpadají tak problémy s konverzí z jednoho kódování do druhého.

Vysoký informační obsah

Pomocí XML značek (tagů) vyznačujeme v dokumentu význam jednotlivých částí textu. Dokumenty tak obsahují více informací, než kdyby se používalo značkování zaměřené na prezentaci (vzhled) – definice písma, odsazení a podobně. XML dokumenty jsou informačně bohatší. To lze samozřejmě s výhodou využít v mnoha oblastech. Největší přínos bude samozřejmě pro prohledávání, kdy můžeme určit i jaký význam má mít hledaný text.

Snadná konverze do jiných formátů

Při používání XML dokumentu potřebujeme také dokument zobrazit. XML samo o sobě žádné prostředky pro definici vzhledu nenabízí. Existuje ale několik stylových jazyků, které umožňují definovat, jak se mají jednotlivé elementy zobrazit. Souboru pravidel nebo příkazů, které definují, jak se dokument převede do jiného formátu, se říká styl.

Jeden vytvořený styl můžeme aplikovat na mnoho dokumentů stejného typu, stejně tak můžeme na jeden dokument aplikovat několik různých stylů. Výsledkem může být např. postscriptový soubor, HTML kód nebo XML s obsahem původního dokumentu.

Stylových jazyků existuje dnes několik. Mezi nejznámější patří asi kaskádové styly (CSS). Ty lze použít pouze pro jednoduché formátování, které dobře poslouží pro zobrazení

dokumentu na obrazovce. Dalším jazykem je XSL (eXtensible Stylesheet Language). Ten umožňuje dokument různě upravovat a transformovat - vybírat části dokumentu nebo generovat obsahy a rejstříky.

Automatická kontrola struktury dokumentu

XML neobsahuje předdefinované značky (tagy), je třeba definovat vlastní značky, které budeme používat. Tyto značky je možné (nepovinně) definovat v souboru DTD (Document Type Definition). Potom je možné automaticky kontrolovat, zda vytvářený XML dokument odpovídá této definici. Program, který tyto kontroly provádí, se nazývá parser. Při vývoji aplikací můžeme parser použít, a ten za nás detekuje většinu chyb v datech.

DTD není jediný definiční jazyk pro XML. Neobsahuje možnost kontrolovat typy dat (čísla, měnové údaje, údaje o datu a čase). To je vlastnost, která chybí při zpracování dat databázového charakteru. V současné době se pod názvem XML schémata pracuje na půdě konsorcia W3C na vytvoření jednotného standardu, který tyto kontroly umožní.

Pro různé standardní aplikace byla postupně vytvořena schémata, která definují značky (názvy elementů) pro konkrétní typy dokumentů. Příkladem může být DocBook, který definuje struktury pro vytváření knih, článků, vědeckých publikací a podobně. Výhodou takových aplikací je, že současně s definičními soubory DTD je dodávána sada stylů (XSL souborů) pro následné zpracování a přípravu pro tisk, nebo pro převod do jiných standardních tvarů (PostScript, HTML atd.).

Další vlastností XML je, že v jednom dokumentu můžeme používat najednou nezávisle na sobě několik druhů značkování pomocí jmenných prostorů (namespaces). To umožňuje kombinovat v jednom dokumentu několik různých definic ve formě DTD nebo schémat bez konfliktů v pojmenování elementů.

Hypertext a odkazy

XML stejně jako HTML umožňuje vytváření odkazů v rámci jednoho dokumentu i mezi dokumenty, má však více možností. Je možné vytvářet i vícesměrné odkazy, které spojují více dokumentů dohromady. Tvorba odkazů je popsána ve třech standardech – XLink, XPointer a XPath.

XPath (XML Path Language) je jazyk, který umožňuje adresovat jednotlivé části dokumentu.

XPointer (XML Pointer Language). Je rozšířením XPath. Používá k určování jednotlivých částí dokumentu ve stylu: „zajímá mě první odstavec třetí kapitoly“. Není nutné ty části dokumentu, na které chceme odkazovat, explicitně označovat pomocí návěstí jako v HTML.

XLink (XML Linking Language) je samotný jazyk pro tvorbu odkazů. Jednotlivé dokumenty se určují pomocí jejich URL adresy, za kterou lze uvést ještě XPointer pro přesnější určení části dokumentu.

2.5.2 Syntaxe XML

XML dokument je text, obvykle Unicode kódovaný jako UTF-8, ale jsou přípustná i jiná kódování.

Na rozdíl od např. HTML, efektivita XML je silně závislá na struktuře, obsahu a integritě. Aby byl dokument považován za správně strukturovaný ("well-formed"), musí mít nejméně následující vlastnosti:

Musí mít právě jeden kořenový (root) element.

Neprázdné elementy musí být ohraničeny startovací a ukončovací značkou. Prázdné elementy mohou být označeny tagem „prázdný element“.

Všechny hodnoty atributů musí být uzavřeny v uvozovkách - jednoduchých (') nebo dvojitých ("), ale jednoduchá uvozovka musí být uzavřena jednoduchou a dvojitá dvojitou. Opačný pár uvozovek může být použit uvnitř hodnot.

Elementy mohou být vnořeny, ale nemohou se překrývat; to znamená, že každý (ne kořenový) element musí být celý obsažen v jiném elementu. Jména elementů v XML rozlišují malá a velká písmena.

Jednoduchý recept v XML jako příklad by mohl vypadat takto:

```
<?xml version="1.0" encoding="UTF-8"?>
<recept jméno="chleba" čas_přípravy="5 minut" čas_vaření="3 hodiny" >
  <titulek>Základní chleba</titulek>
```

Mouka

Kvasnice

Horká voda

Sůl

```
<instrukce>
  <krok>Smíchejte všechny přísady dohromady a dobře prohněťte.</krok>
  <krok>Zakryjte tkaninou a nechejte hodinu v teplé místnosti.</krok>
  <krok>Znovu prohněťte, umístěte na plech a pečte v troubě.</krok>
</instrukce>
</recept>
```

2.5.3 Aplikace XML

Příklady aplikace XML:

XHTML – Nástupce jazyka HTML.

RDF – Resource Description Framework, specifikace, která umožňuje popsat metadata, např. obsah a anotace HTML stránky.

SMIL – Synchronized Multimedia Integration Language, popisuje multimedia pomocí XML.

MathML – Mathematical Markup Language je značkovací jazyk pro popis matematických vzorců a symbolů pro použití na webu.

SVG – Scalable Vector Graphics je jazyk pro popis dvourozměrné vektorové grafiky, statické i dynamické (animace).

DocBook – Sada definic dokumentů a stylů pro publikační činnost.

Jabber – Protokol pro Instant messaging .

2.5.4 Verze XML

Aktuální verze XML je 1.1 (od 4.února 2004). První verze XML 1.0 existuje ve třetí revizi. Obě verze se liší v požadavcích na použité znaky v názvech elementů, atributů atd. Verze 1.0 dovozovala pouze užívání znaků platných ve verzi Unicode 2.0, která obsahuje většinu světových písem, ale neobsahuje později přidané sady jako je Mongolština a podobně. Verze XML 1.1 zakazuje pouze řídicí znaky, což znamená, že mohou být použity jakékoli jiné znaky.

Je třeba poznamenat, že omezení ve verzi 1.0 se vztahuje pouze na názvy elementů a atributů. Jinak obě verze dovolují v obsahu dokumentu jakékoli znaky. Verze 1.1 je tedy nutná, pokud potřebujeme psát názvy elementů v jazyku, který byl přidán do Unicode později.

Další menší změna mezi XML 1.0 a 1.1 je, že řídicí znaky se nyní mohou vkládat jen jako escape sekvence, a se speciálními znaky „form-feed“ se musí zacházet jako s bílými znaky.

Všechny dokumenty verze 1.0 budou platné ve verzi 1.1 s jednou výjimkou: dokumenty deklarované s kódováním ISO-8859-1, které jsou ve skutečnosti v kódu Windows CP1252, nemusejí být nyní platné. Důvodem je, že CP1252 používají blok řídicích znaků pro speciální zobrazení, jako jsou znaky €, Œ, and ™. Dokumenty s deklarací CP1252 zůstávají platné. [1]

2.6 HTTP

HTTP (Hyper Text Transfer Protocol) je Internetový protokol určený původně pro výměnu hypertextových dokumentů ve formátu HTML. Používá obvykle port TCP/80, verze 1.1 protokolu je definována v RFC 2616. Tento protokol je spolu s elektronickou poštou tím nejvíce používaným a zasloužil se o obrovský rozmach internetu v posledních letech.

V současné době je používán i pro přenos dalších informací. Pomocí rozšíření MIME umí přenášet jakýkoli soubor (podobně jako e-mail), používá se společně s formátem XML pro tzv. webové služby (spouštění vzdálených aplikací) a pomocí aplikačních bran zpřístupňuje i další protokoly, jako je např. FTP nebo SMTP.

HTTP používá jako některé další aplikace tzv. jednotný lokátor prostředků (URL, Uniform Resource Locator), který specifikuje jednoznačné umístění nějakého zdroje v Internetu.

K protokolu HTTP existuje také jeho bezpečnější verze HTTPS, která umožňuje přenášená data šifrovat a tím chránit před odposlechem či jiným narušením.

2.6.1 Činnost protokolu

Protokol funguje způsobem dotaz-odpověď. Uživatel (pomocí programu, obvykle internetového prohlížeče) pošle serveru dotaz ve formě čistého textu, obsahujícího označení požadovaného dokumentu, informace o schopnostech prohlížeče apod. Server poté odpoví pomocí několika řádků textu popisujících výsledek dotazu (zda se dokument poda-

řilo najít, jakého typu dokument je atd.), za kterými následují data samotného požadovaného dokumentu.

Pokud uživatel bude mít po chvíli další dotaz na stejný server (např. proto, že uživatel v dokumentu kliknul na hypertextový odkaz), bude se jednat o další, nezávislý dotaz a odpověď. Z hlediska serveru nelze poznat, jestli tento druhý dotaz jakkoli souvisí s předchozím. Kvůli této vlastnosti se protokolu HTTP říká bezstavový protokol – protokol neumí uchovávat stav komunikace, dotazy spolu nemají souvislost. Tato vlastnost je nepříjemná pro implementaci složitějších procesů přes HTTP (např. internetový obchod potřebuje uchovávat informaci o identitě zákazníka, o obsahu jeho „nákupního košíku“ apod.). K tomuto účelu byl protokol HTTP rozšířen o tzv. HTTP cookies, které umožňují serveru uchovávat si informace o stavu spojení na počítači uživatele.

2.6.2 Ukázka komunikace

Uživatelský program se připojí na server cs.wikipedia.org a zašle následující text:

```
GET /wiki/Wikipedie HTTP/1.1
Host: cs.wikipedia.org
User-Agent: Mozilla/5.0 Gecko/20040803 Firefox/0.9.3
Accept-Charset: UTF-8,*
```

Tímto dotazem žádá o dokument /wiki/Wikipedie na serveru cs.wikipedia.org, sděluje svou totožnost (Mozilla Firefox verze 0.9.3) a oznamuje, že podporuje kódování UTF-8. (Ve skutečném dotazu je podobných informací ještě více, toto je zjednodušený příklad.)

Server pak odpoví:

```
HTTP/1.0 200 OK
Date: Fri, 15 Oct 2004 08:20:25 GMT
Server: Apache/1.3.29 (Unix) PHP/4.3.8
X-Powered-By: PHP/4.3.8
Vary: Accept-Encoding, Cookie
Cache-Control: private, s-maxage=0, max-age=0, must-revalidate
Content-Language: cs
Content-Type: text/html; charset=utf-8
```

Za touto hlavičkou následuje jeden prázdný řádek (označující její konec) a pak požadovaný HTML dokument. Hlavička obsahuje informaci o tom, že dotaz se podařil (první řádek:

„200 OK“), datum a čas vyřízení dotazu, popis serveru, který odpovídá, informace o typu vráceného dokumentu (MIME typ text/html v kódování UTF-8) a další informace.

2.6.3 Dotazovací metody

HTTP definuje několik metod, které se mají provést nad uvedeným objektem (dokumentem). <metoda> <objekt> HTTP/<verze>

GET - Požadavek na uvedený objekt. Je to nejpoužívanější metoda. Je používána když se podíváte ráno na zpravodajský i-magazín, přes den stahujete RSS a nebo stahujete novou verzi webového prohlížeče z webu.

HEAD - To samé jako metoda GET, ale už nepředává data. Poskytne pouze metadata o požadovaném cíli (velikost, typ, datum změny, ...).

POST - Odesílá uživatelská data na server. Používá se například při odesílání formuláře na webu. S předaným objektem se pak zachází podobně jako při metodě GET.

PUT - Nahraje data na server. Objekt je jméno vytvářeného souboru. Používá se například při vývoji webových stránek.

DELETE - Smaže uvedený objekt ze serveru. Jsou na to potřeba jistá oprávnění stejně jako u metody PUT.

TRACE

OPTIONS - Dotaz na server, jaké podporuje metody.

CONNECT - Spojí se s uvedeným objektem před uvedený port. Používá se při průchodu skrze proxy pro ustanovení kanálu SSL. [1]

II. PRAKTICKÁ ČÁST

3 SEZNÁMENÍ S PRACÍ V GNU OCTAVE

3.1 Instalace

První věcí, kterou je třeba udělat, pokud chceme pracovat s tímto softwarem, je instalace samotného produktu. Vzhledem k tomu, že Octave vznikl jako open-source je tedy volně stažitelný z internetu a není problém jej nalézt v mnoha mutacích pro různé platformy. Z těch základních vyjmenujme např. Unix (Linux), Mac OS či MS Windows. Jelikož nejrozšířenějším operačním systémem je OS MS Windows od firmy Microsoft (a stejně tak i já jsem tento program provozoval na Microsoft Windows XP Home Edition) - budu se věnovat verzi pod tímto systémem.

3.2 Spouštění a ukončení

Po úspěšné instalaci můžete program spustit buď z plochy, kam se při instalaci umístil zástupce nebo přes nabídku *Start* a položku *GNU Octave 2.1.50* (resp. č. verze, kterou používáte). Pokud je vše správně, program zobrazí inicializační zprávu a poté již příkazový řádek indikuje, že je připraven přijímat vstup - objeví se příkazová řádka. Poté můžete ihned začít zadávat jednotlivé Octave příkazy. Okno se spuštěným programem vypadá přibližně (závisí na nastavení vzhledu systému) takto:



```
GNU Octave, version 2.1.50 (i686-pc-cygwin).
Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003 John W. Eaton.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Report bugs to <bug-octave@bevo.che.wisc.edu>.

>> █
```

Obr. 4 Okno s vlastním programem

Pokud se při práci s programem dostanete do jakéhokoli problému, jednoduše se veškerá aktuální činnost programu dá zastavit kombinací kláves CTRL a C. Po jejím stisknutí se veškerá aktivita přeruší a program se vrátí na příkazový řádek.

Ukončení samotného programu se provádí buď příkazů `quit` nebo `exit` nebo jednoduchým stisknutím *Windows* tlačítka pro zavření okna v pravém horním rohu.

3.3 Práce s programem

Program po spuštění chvíli inicializuje své vnitřní zabudované proměnné, ale po krátké prodlevě je připraven. Na výše uvedeném obrázku je možné vidět příkazový řádek očekávající příkazy. Je tedy možné ihned po zobrazení blikajícího černého kurzoru začít programovat. Jednou z možností jak s Octave pracovat je přímé zadávání jednotlivých příkazů, což je vhodné pro dílčí výpočty či jednoduché funkce. Pokud však máme v úmyslu pracovat se složitými funkcemi a rozsáhlou databází datových struktur a proměnných je vhodné použít externích souborů, které do programu můžeme kdykoli nahrát. Ne, že by vnitřní prostředí nebylo přehledné, ale pokud je příkazů příliš a najde se nějaká chyba je mnohem snazší a uživatelsky příjemnější ji najít v externím textovém souboru, opravit ji a práci si uložit pro pozdější použití. Další výhodou je jednoduchá opakovatelnost volání. Po ukončení programu totiž veškeré neuložené data ztrácíte, stejně jako obsahy proměnných.

4 UKÁZKOVÉ PŘÍKLADY DEMONSTRUJÍCÍ ZÁKLADNÍ MOŽNOSTI PROGRAMU

Program Octave je volně dostupnou variantou pro komerční Matlab, jehož název vznikl zkrácením slov matrix a laboratory. Stejně jako Matlab je tedy jazyk Octave postaven na maticích. Všechny operace jsou maticové, jen někdy jsou tyto matice jednoprvkové. Tyto matice mohou mít jak reálné, tak i imaginární členy. Proto přímo v prostředí programu zde uvedu základní práci s maticemi, ale také s věcmi, které považuji za základní pro orientaci ve struktuře příkazů a celkově pro pochopení smyslu celého jazyka.

4.1 Vytvoření matice

Pro vytvoření matice a její následné uložení v proměnné tak, aby s ní bylo možno později pracovat lze provést příkazem:

```
a = [ 1, 1, 2; 3, 5, 8; 13, 21, 34 ]
```

Octave odpoví výpisem matice v úhledně zarovnaných sloupcích:

```
>> a=[1,1,2;3,5,8;13,21,34]
a =
      1      1      2
      3      5      8
     13     21     34
>> █
```

Obr. 5 Výpis matice

Pokud je však příkaz končen středníkem, výsledek není vypsán.

Např. :

```
b = rand (3, 2);
```

Má za následek provedení příkazu bez jakéhokoli výpisu:

```
>> b=rand(3,2);
>> █
```

Obr. 6 Omezený výpis matice

4.2 Výpis proměnné

K zobrazení hodnoty proměnné jakéhokoli typu stačí jednoduše napsat její název. Například zobrazení hodnot uložených v matici `b` (definována výše) provedeme následovně:

```
>> b
b =
    0.14181    0.89678
    0.22955    0.87483
    0.01485    0.21186
>> █
```

Obr. 7 Výpis proměnné

4.3 Početní operace

Octave lze využít pro základní výpočetní operace. Pro psaní vzorců se používají znaky `+`, `-`, `*`, `/`, `^`, kulaté závorky apod., stejně jako na běžné kalkulačce. Dodejme pro úplnost, že Octave patří mezi "lepší kalkulačky", kterým nedělá problém přednost násobení před sčítáním. Taktéž lze použít standardní operátory (známé např. z jazyka C) `&`, `&&`, `|`, `||`, `--`, `++`. O kvalitách tohoto programu také jistě svědčí fakt, že jazyk umožňuje též numerickou integraci soustav jednoduchých diferenciálních rovnic nebo soustav jednoduchých diferenciálních a algebraických rovnic nemluvě o vykreslování grafů i 3D drátěných modelů.

Octave používá tedy pohodlný zápis operátorů k provádění operací nejen s maticemi. Například pro násobení matice skalární hodnotou napíšeme příkaz:

```
2 * a
```

A finálním výstupem bude přehledná tabulka výsledných hodnot matice:

```
>> 2*a
ans =
     2     2     4
     6    10    16
    26    42    68
>> █
```

Obr. 8 Výpis násobení matice skalární hodnotou

Poznamenejme, že výsledek, který není uložen do žádné z proměnných se ukládá do automaticky vzniklé zabudované proměnné `ans`, která v sobě výsledek uchová až do další početní operace, kdy se pak její hodnota přepíše výsledkem novým.

Praktickou pomůckou může také být řešení soustav o n neznámých, např. zadání:

Řešte soustavu lineárních rovnic (určete hodnoty x_1, x_2, x_3):

$$4x_1 + 6x_2 + 8x_3 = 40$$

$$14x_1 + 4x_2 + 2x_3 = 28$$

$$2x_1 + 10x_2 + 16x_3 = 70$$

Řešení se v programu Octave provede následovně:

```
A=[4 6 8;14 4 2;2 10 16]
```

```
B=[40;28;70]
```

```
A\B
```

4.4 Vykreslování grafů

Všechny vykreslovací funkce Octave využívají *gnuplot* k zajištění grafiky. Většina typů grafů může být generována pomocí vykreslovacích funkcí, které jsou známé z Matlabu.

4.4.1 2D grafy

Jejich realizace se provádí pomocí příkazu `plot` následovaného povinnými či nepovinnými argumenty, které ovlivňují vzhledovou část. Ve výchozím nastavení každé volání tohoto příkazu inicializuje výstupní zařízení, tj. smaže případný obsah z předchozích kreslicích výstupů. Je však možnost, buď pomocí funkce `hold` podržet aktuální obsah výstupního zařízení nebo přímo příkazu `plot` dát všechny funkce, které chceme vykreslit v argumentu.

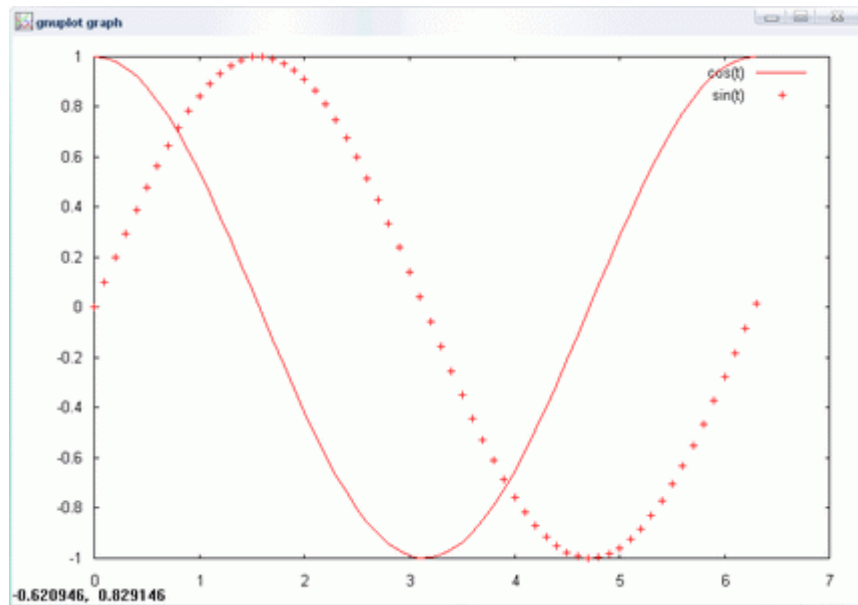
Ukázkou by mohl být následující příklad:

```
t=0:0.1:6.3;
```

```
plot(t,cos(t), "-;cos(t);",t,sin(t), "+;sin(t);");
```

kde proměnná t obsahuje po provedení 64 hodnot následující: od 0 do 6,3 po kroku o velikosti 0,1. Dále pak následuje druhý příkaz k samotnému vykreslení grafu funkce sinus a kosinus. Použijeme intervalu definovaného t – tedy od 0 do 2π , dalšími argumenty volí-

me styl čáry grafu a také popisek – legendu vpravo nahoře. Výstup pak vypadá následovně:



Obr. 9 Vykreslený graf funkcí sinus a cosinus

Dodejme, že Octave disponuje speciálními předdefinovanými grafy, které se používají pomocí funkcí k jejich vykreslování určených. Jedná se např. o grafy sloupcové, s logaritmickými osami, schodové, chybové nebo histogramy. Více o nich se dozvíte v příloze této práce.

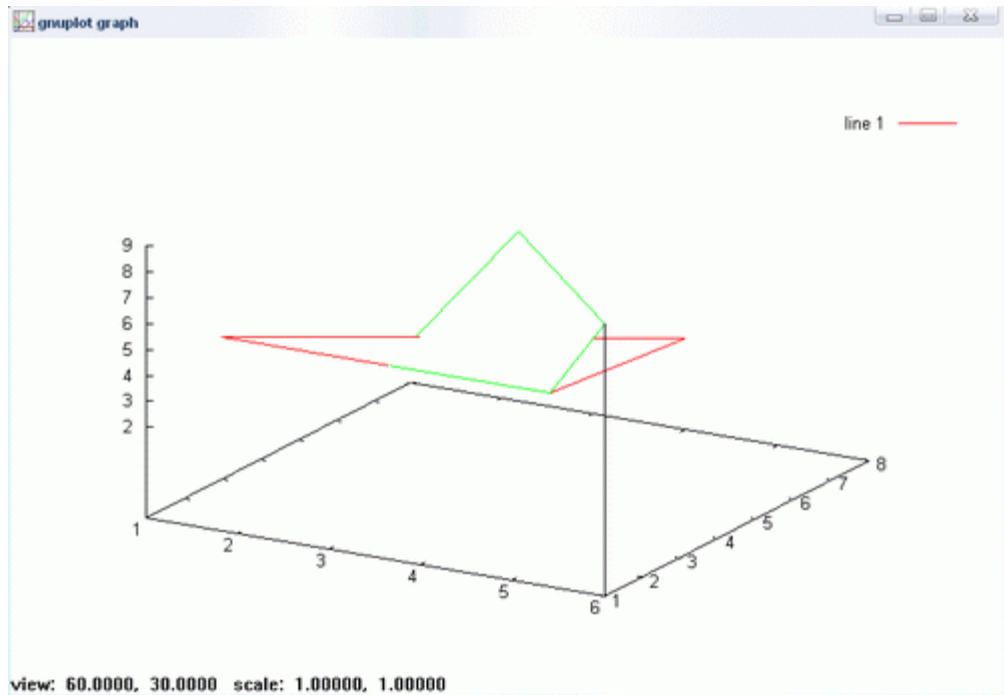
4.4.2 3D grafy

Třírozměrný grafický výstup Octave reprezentují jednoduché drátěné modely a funkce mesh.

Například zadáme-li

```
x - řádkový vektor (délka 3)
y - řádkový vektor (délka 3)
z - matice 3x2
mesh (x, y, z)
```

Tak výstupem bude zobrazení třírozměrné plochy:

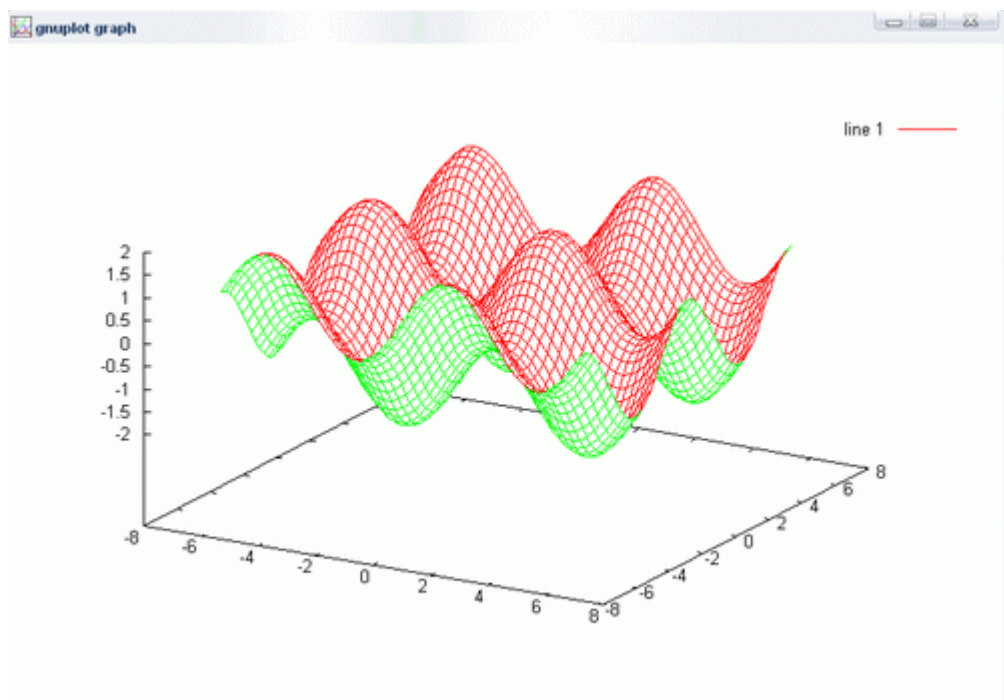


Obr. 10 Vykreslený 3D graf

Estetičtější laděného grafu můžeme docílit například sadou příkazů:

```
x=y=linspace(-2*pi,2*pi,50);  
[xx,yy]=meshgrid(x,y);  
mesh(x,y,sin(xx)+cos(yy))
```

kde pak výstup vypadá následovně:



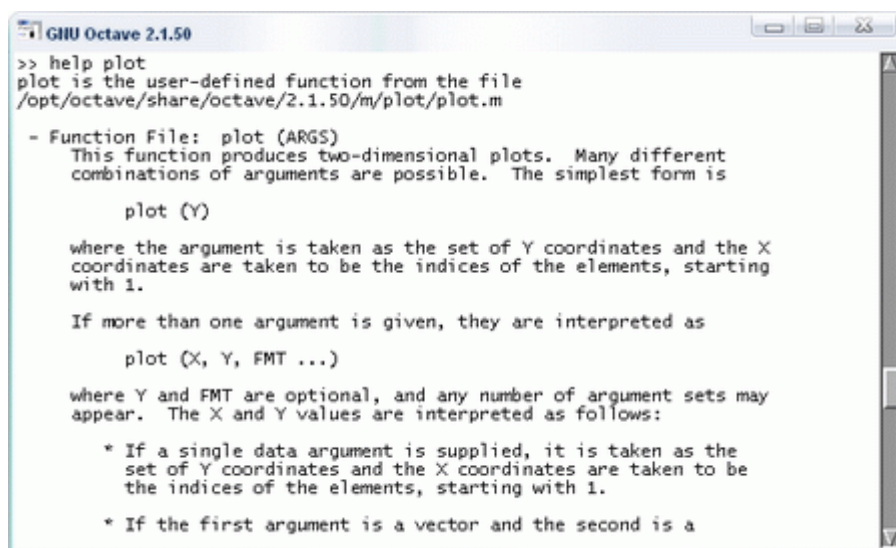
Obr. 11 Komplikovanější 3D vykreslený objekt

4.5 Nápověda

Program má také velice propracovanou nápovědu k jednotlivým příkazům. Jednoduše stačí na příkazovém řádku spustit příkaz `help` s parametrem funkce, ke které chcete nápovědu získat. Chceme-li získat podrobnější informace o používání funkce k vykreslování grafů `plot` není nic jednoduššího, než na příkazovou řádku napsat:

```
help plot
```

a dočkáme se výstupu, ve kterém Octave vypíše veškeré dostupné informace, které k danému argumentu – funkci má:



```
GNU Octave 2.1.50
>> help plot
plot is the user-defined function from the file
/opt/octave/share/octave/2.1.50/m/plot/plot.m

- Function File: plot (ARGS)
  This function produces two-dimensional plots. Many different
  combinations of arguments are possible. The simplest form is

  plot (Y)

  where the argument is taken as the set of Y coordinates and the X
  coordinates are taken to be the indices of the elements, starting
  with 1.

  If more than one argument is given, they are interpreted as

  plot (X, Y, FMT ...)

  where Y and FMT are optional, and any number of argument sets may
  appear. The X and Y values are interpreted as follows:

  * If a single data argument is supplied, it is taken as the
  set of Y coordinates and the X coordinates are taken to be
  the indices of the elements, starting with 1.

  * If the first argument is a vector and the second is a
```

Obr. 12 Výpis nápovědy programu

5 TVORBA HTML REFERENČNÍ PŘÍRUČKY

5.1 Příprava

5.1.1 Programové možnosti tvorby internetových stránek

Při tvorbě www stránek se lze ubírat dvěma směry. První bych zmínil ten méně používaný. Dá se říci amatérský, avšak pro běžného uživatele, který o této problematice nemá příliš informací a nebo jen prostě nechce věnovat tvorbě stránky příliš času, dostačující. Cenou je však neefektivita takto vytvořených stránek, pokulhávající design a možnosti, které stránky nabízejí jako celek. Řeč je o nástrojích typu Microsoft FrontPage. Tento a mu podobné

nástroje umožňují uživateli stránku tvořit intuitivně bez jakýchkoli znalostí o zdrojových kódech HTML. Uživatel pouze píše text, vkládá obrázky, odkazy a výsledek se za něj uloží do požadovaného formátu. Jak již bylo zmíněno, tento způsob tvorby stránek je neefektivní a doba načítání takovéto stránky je při větším množství dat opravdu pozorovatelná. Dá se také využít možnosti tvorby vlastních stránek přímo v prostředí vyspělejšího textového editoru – např. Microsoft Word, kde výslednou práci stačí jednoduše uložit jako internetovou stránku. Program se pak sám postará o převod.

Druhou možností je cesta tzv. vlastnoručního programování – psaní kódu. Člověk, který již ví, k čemu je jaká značka, tag apod. nemá problém vytvořit internetovou stránku v libovolném textovém editoru (poznámkový blok aj.). Pro tvorbu stránek je ale vhodné použít vyspělých nástrojů k tomu určených, kterých je na internetu volně ke stažení celá řada. Výhod, které tento způsob přináší je spousta. Oproti poznámkovému bloku je například zdrojový kód barevně oddělen. Značky mají jinou barvu jako text, obrázky či odkazy. Programátor se pak lépe orientuje v tom, co vlastně píše a při úpravách se snadno zorientuje. Také je většinou k dispozici náhled, který zobrazí aktuální stránku stejně, jako internetový prohlížeč. Vyspělejší nástroje dokáží simulovat chování a překlad jednotlivých prohlížečů – MS Internet Explorer, Opera, Mozilla, ...

5.1.2 Výběr softwaru

Na základě mých předchozích zkušeností s vytvářením www stránek jsem si vybral TSW Webcoder verze 5. Při dřívější politice firmy Kasper Tanggaard byl tento software

zdarma, jen „obtěžoval“ prosbou o registraci. Tato registrace byla pro firmy zpoplatněna – aby mohl být financovaný další vývoj programu, ale pro studenty byla registrace zdarma. Nyní však již firma vydala novou verzi tohoto programu a změnila svou licenční politiku. Nadále je ale možné tento program volně používat, ale registrace zdarma již není možná. Proto jediným omezením tohoto programu je sedmi sekundový odpočet při prosbě o registraci. Použil jsem pro vytvoření přílohy této práce neregistrovanou verzi.

Oficiální logo programu s informacemi o verzi, internetových stránkách a výrobci:

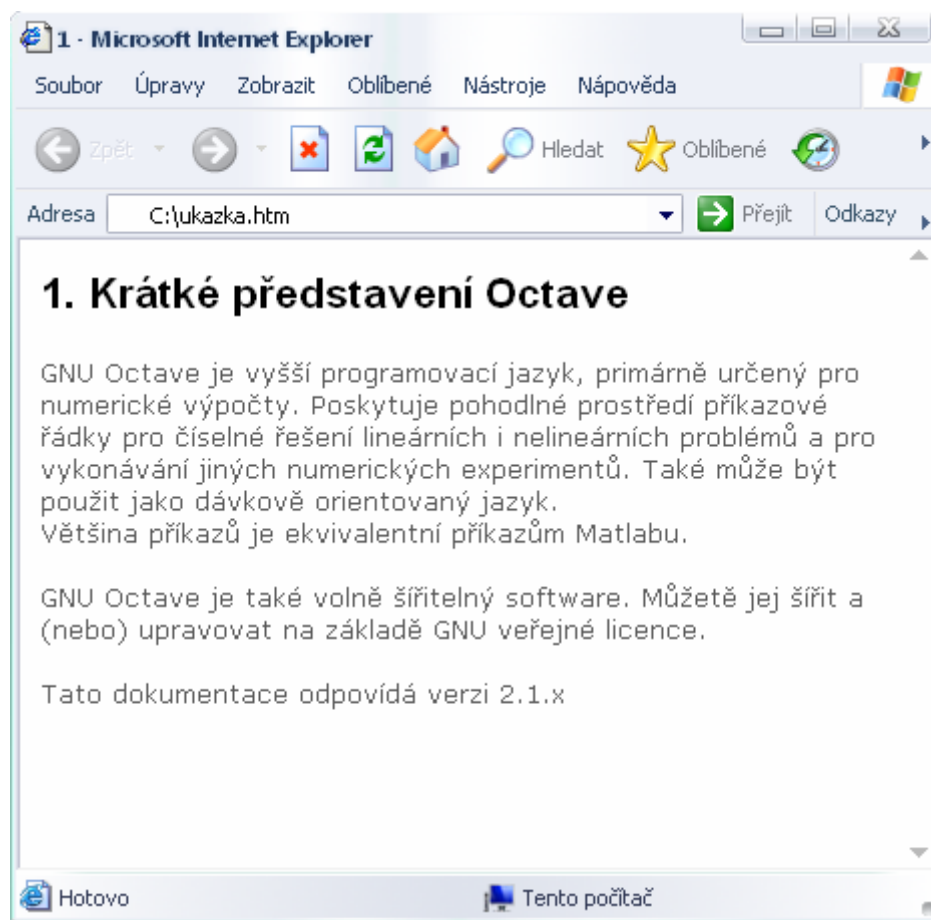


Obr. 13 Informace o programu TSW Webcoder

5.1.3 Srovnání efektivity vybraných nástrojů

Pro srovnání efektivity výše zmíněných způsobů tvorby www stránek bych uvedl zdrojový kód stránky, který jednou vygeneruje aplikace Microsoft Word při možnosti volby *Soubor – Uložit jako – Webová stránka (*.htm, *.html)* a jako druhý kód, který je psán přímo v prostředí TSW Webcoder. Toto prostředí však není podmínkou, stejně efektivní kód lze vytvořit v libovolném jednoduchém textovém editoru.

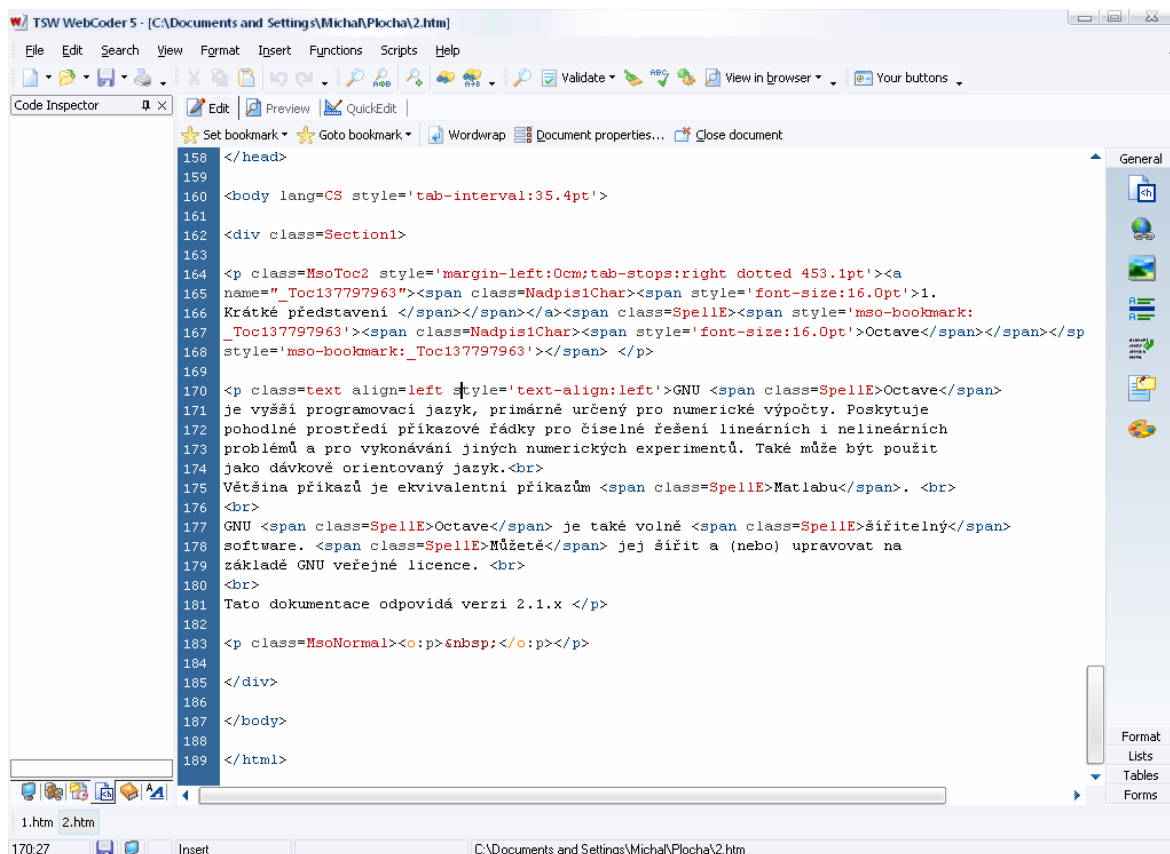
Stránka, na které bude efektivita předváděna vypadá při své reprezentaci v internetovém prohlížeči následovně:



Obr. 14 Efektivita: Zdrojová stránka

Microsoft Word

Po vložení požadovaného textu do aplikace a jejím následném uložení ve formátu webové stránky bylo vygenerováno 189 řádků zdrojového kódu obsahující všemožné definice stylů a přebytečné části kódu. Stránka je funkční, vypadá jak má, ale pro uživatele, který si ji chce z internetu stáhnout a přečíst znamená časové zdržení již při samotném stahování stránky, protože je stránka mnohonásobně větší, tak při její reprezentaci v prohlížeči, kdy se musí zpracovat přebytečný kód.



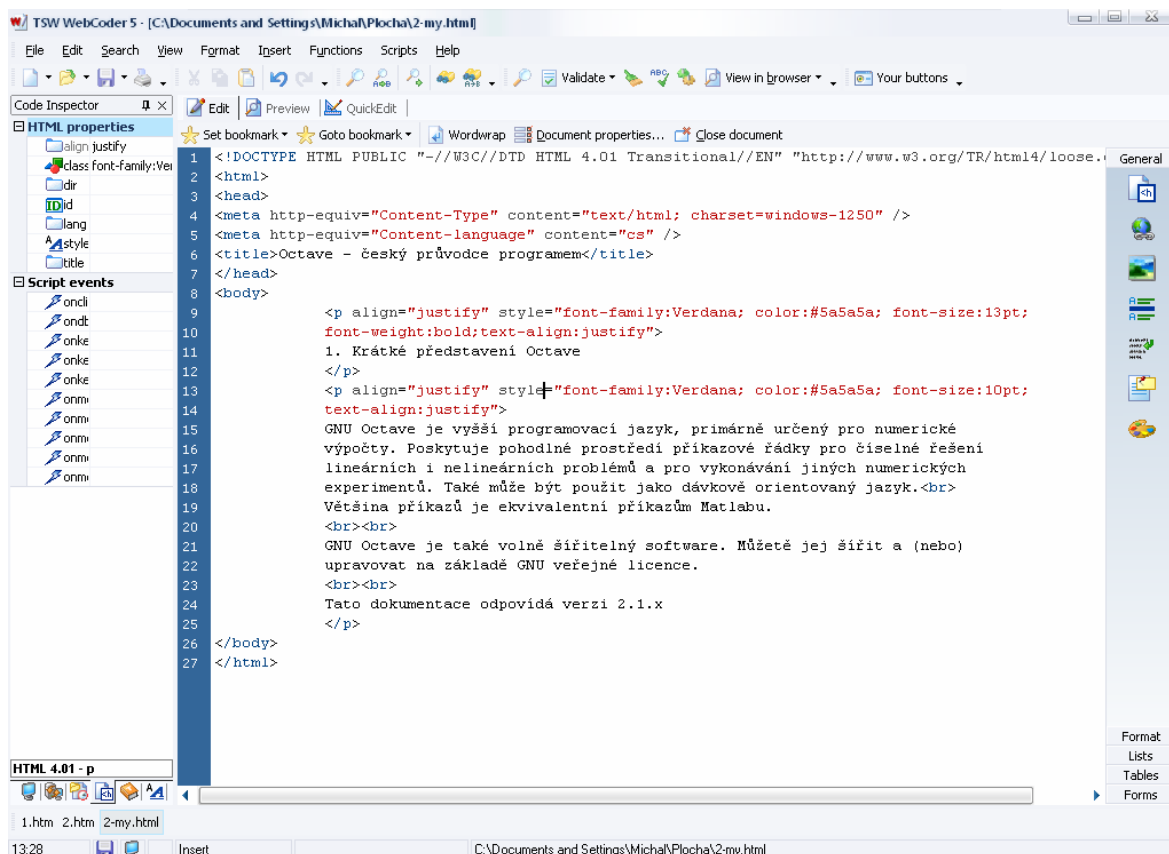
```
158 </head>
159
160 <body lang=CS style='tab-interval:35.4pt'>
161
162 <div class=Section1>
163
164 <p class=MsoToc2 style='margin-left:0cm;tab-stops:right dotted 453.1pt'><a
165 name="_Toc137797963"><span class=Nadpis1Char><span style='font-size:16.0pt'>1.
166 Krátké představení </span></span></a><span class=SpellE><span style='mso-bookmark:
167 _Toc137797963'><span class=Nadpis1Char><span style='font-size:16.0pt'>Octave</span></span></sp
168 style='mso-bookmark:_Toc137797963'></span> </p>
169
170 <p class=text align=left style='text-align:left'>GNU <span class=SpellE>Octave</span>
171 je vyšší programovací jazyk, primárně určený pro numerické výpočty. Poskytuje
172 pohodlné prostředí příkazové řádky pro číselné řešení lineárních i nelineárních
173 problémů a pro vykonávání jiných numerických experimentů. Také může být použit
174 jako dávkové orientovaný jazyk.<br>
175 Většina příkazů je ekvivalentní příkazům <span class=SpellE>Matlabu</span>. <br>
176 <br>
177 GNU <span class=SpellE>Octave</span> je také volně <span class=SpellE>šířitelný</span>
178 software. <span class=SpellE>Můžete</span> jej šířit a (nebo) upravovat na
179 základě GNU veřejné licence. <br>
180 <br>
181 Tato dokumentace odpovídá verzi 2.1.x </p>
182
183 <p class=MsoNormal><o:p>&nbsp;</o:p></p>
184
185 </div>
186
187 </body>
188
189 </html>
```

Obr. 15 Efektivita: MS Word

Vlastnoručně psaná stránka

Stejná stránka, vlastnoručně napsaná v HTML kódu za použití CSS stylů zabírající přesně 27 řádků. Rozdíl ve velikosti:

- MS Word – 5,8kB (čistého textu)
- Vlastnoručně – 1,2kB (čistého textu)



Obr. 16 Efektivita: Vlastnoručně psaná stránka

5.2 Vlastní práce na HTML referenční příručce

Jako cenný zdroj informací jsem použil internetové stránky s originální dokumentací k programu GNU Octave, které jsem se jal překládat. K nalezení jsou na stránce:

<http://www.gnu.org/software/octave/doc/interpreter/> [11.06.2006]

Má snaha nebyl pouhý a strohý doslovný překlad, ale srozumitelný manuál, který by měl zajišťovat potencionálnímu zájemci o tuto problematiku informace ať už z hlediska kvantitativního či kvalitativního. Mnohdy však překlad byl opravdovým oříškem, zejména v částech, kde matematické a algebraické pojmy zastávaly většinu slov ve větách. A tak s pomocí Seznam.cz online slovníku:

<http://slovník.seznam.cz> [11.06.2006]

a anglického výkladového slovníku Merriam Webster OnLine:

<http://www.m-w.com> [11.06.2006]

jsem se snažil o co nejpřesnější překlad.

5.2.1 Umístění na internetu

Vytvořené stránky jsou umístěny na nově vytvořené a zaregistrované doménové adrese na serveru webzdarma.cz. Jsou tak pro kohokoli volně dostupné z internetu na adrese:

<http://octave.wz.cz>

5.2.2 Počátek práce

Mou snahou byla co největší obsahová správnost překladu a hlavně přehlednost stránek, snadná orientace a moderněji laděný design. Grafické prvky jsou mého vlastního návrhu, stejně jako celá koncepce stránek.

Jako první bylo třeba zvolit samotný způsob tvorby stránek. Jelikož jsem prošel kurzem tvorby www stránek, zvolil jsem profesionálnější cestu a kód si psal sám. O výběru nástroje jsem se zmínil v kapitole 5.1.2 Výběr softwaru.

5.2.3 Grafické rozvržení

Další na řadu přišlo samotné grafické ztvárnění, kostra stránky a samotné navigační menu. Jako pozadí jsem zvolil vlastní namíchanou modrou barvu s šedými proužky. Pro ušetření místa na serveru, ale také hlavně ušetření času při načítání pozadí stránky, jsem zvolil miniaturní velikost obrázku – 7x7 pixelů motivu (Obr. 17) – který se opakuje po celé ploše stránky.

Pro ukázkou velikostně vypadá motiv pozadí takto:



Obr. 17 Pozadí stránky

Poté, co stránka dostala modro-bílý nádech z pozadí a samotné tabulky bylo třeba vymyslet barvu vnitřního loga a samozřejmě textu, kterého je nejvíce. Po několika pokusech jsem jako nejvhodnější barevnou kombinaci zvolil oranžovou pro oživující grafické prvky a tmavě šedou pro text. Výraznou oranžovou pak doplňují barvy, ze kterých je vlastně namíchaná a to konkrétně červená a žlutá.

Návrh loga a jeho realizace byla provedena v externím grafickém programu, stejně tak jako nadpisy hlavních částí stránek, k jejichž tvorbě bylo užito speciálního fontu TRAK.

5.2.4 Koncepce stránek

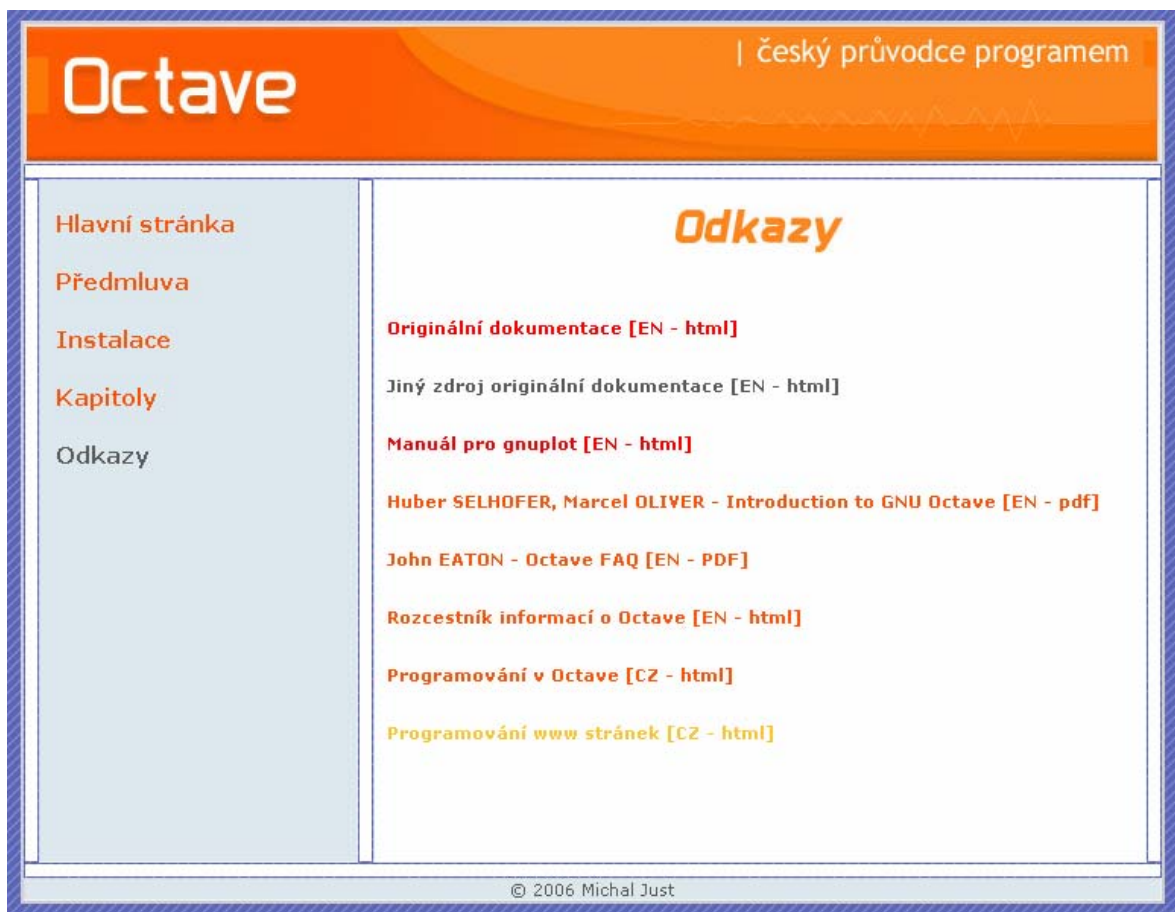
Celá stránka je pak koncipována jako tabulka, která má šířku 750 pixelů, je vycentrovaná na střed a hranice tabulky jsou skryty. Celkově tabulka obsahuje 5 řádků a 5 sloupců, které jsou funkčně poskládány a pospojovány pro maximální přehlednost.

Pro ukázkou je možné předvést stránku s odkazy, kde byla nastavena šířka hranice pro jednotlivé oddělovače sloupců a řádků na 2 pixely. Je zde vidět celkové rozložení stránky, spojení jednotlivých buněk tabulky a barevný odstín jednotlivých buněk pro barevné odlišení.

Vše má na svědomí následující kousek kódu a hodnota proměnné border:

```
<table style="border:2px solid rgb(207,207,207);" background="#ffffff"
align="center" border="2" cellpadding="0" cellspacing="0" width="750">
```

Stránka v konstrukčním režimu vypadá následovně:



Obr. 18 Rozvržení tabulky

Pokud však výše zmíněný kód upravíme na finální verzi:

```
<table style="border:2px solid rgb(207,207,207);" background="#ffffff"
align="center" border="0" cellpadding="0" cellspacing="0" width="750">
```

kde jednotlivé prvky znamenají:

`<table` – začátek hlavní definice tabulky

`style` – označuje místo, kde se bude vkládat nějaký kaskádový styl. V tomhle případě se jedná o položky:

`border:2px` – šířka vnějšího ohraničení tabulky (2 pixely)

`solid` – styl čáry – plná

`rgb` – barva složená ze tří čísel – složek červené, zelené a modré barvy

`background` – pozadí celé tabulky (bílá)

`align` – zarovnání (na střed)

`border` – šířka ohraničení řádků a sloupců (0 pixelů)

`cellpadding` – vnitřní okraj buněk (0 pixelů)

`cellspacing` – vnější okraj buněk (0 pixelů)

`width` – šířka (750 pixelů)

`>` – ukončovací znak – končí definici tabulky. Dále se může pokračovat libovolným textem či další definicí. Nejčastěji pak definicí řádků a sloupců tabulky.

Stránka pak ve své konečné podobě bude vypadat následovně:



Obr. 19 Finálně upravená tabulka a vzhled stránky

5.2.5 Propojení hypertextovými odkazy, hlavní nabídka

Nejvíce práce, ihned po samotném překladu, zabralo propojení jednotlivých kapitol mezi sebou odkazy. Nejen, že je možné se z kterékoli libovolné kapitoly kliknutím na hlavní logo dostat na hlavní stránku, ale také jsou stránky a kapitoly propojeny mezi sebou. Vše začíná hlavní stránkou (*index.html*), kde je možné si vybrat z nabídky:

- Předmluvu autora programu
- Instalaci, kde je podrobný popis, jak software nainstalovat
- Kapitoly, čímž začíná samotná referenční příručka
- Odkazy, kde je k dispozici seznam praktických internetových stránek k dalšímu podrobnějšímu studiu.

Po kliknutí na část, kde se nachází kapitoly se uživatel dostane na přehled veškerého obsahu, který je k dispozici. Aktuálně je to 22 kapitol s mnohočetným dělením v jednotlivé

podkapitoly až do třetí úrovně. Další případné úrovně již nejsou v seznamu vedeny a figurují pouze jako nadpisy.

Jakmile je otevřena první (nebo kterákoli jiná) kapitola, okamžitě se načte její obsah, zobrazí se a nabídka na levé straně, kde bylo navigační menu změně svou podobou. Aktuální kapitola je zobrazena šedou barvou (a neaktivní jako odkaz), ale v jejím bezprostředním okolí bude seznam všech kapitol plus vypsání všechny podkapitoly aktuální nadřazené kapitoly. To v praxi znamená, že pokud otevřeme třetí kapitolu, v levé části bude seznam všech hlavních 22 kapitol a 3. kapitola bude rozbalená v tom smyslu, že veškeré podkapitoly budou taktéž vypsány. Tím však propojení jednotlivých kapitol nekončí.

Na konci textu, který je k dané problematice vypsán se nachází směrový navigační pruh, ze kterého je možné se dostat na obě strany – o kapitolu zpět, či vpřed. Takhle se dá bez problémů celé příručka pohodlně přečíst. Jakmile se narazí na konec textu, klikne se pro další stránku. Propojeny jsou veškeré kapitoly i podkapitoly v logické návaznosti na sebe.

5.2.6 Ukázkový soubor zdrojového kódu



Obr. 20 Ukázkový soubor při své reprezentaci v prohlížeči


```
<A href="./8-4.html" class=odkaz>&nbsp;<
8.4 Výchozí proměnné prostředí <</A>
</td>
<td align="right">
<A href="./9-1.html" class=odkaz>> 9.1 In-
dexové výrazy >&nbsp;</A>
</td>
</tr>
</table>
</td>
<td bgcolor="#ffffff" width="8">
</td>
</tr>
<tr>
<td bgcolor="#ffffff" colspan="5" height="10">
</td>
</tr>
<tr>
<td bgcolor="#dce8ed" colspan="5">
<div align=center class=spodek>© 2006 Michal Just</div>
</td>
</tr>
</table>
</body>
</html>
```

5.2.7 Statistické údaje

Celkové složení a statistické údaje týkající se HTML verze příručky:

Místo zabírající na disku: **1,85MB**

Počet souborů: **198 v 6 adresářích**

Složení souborů:

- **1x definice kaskádových stylů**
- **83x souborů grafiky - obrázky** (z toho 55 ukázkových příkladů)
- **114x souborů s HTML kódem obsahující překlad**

Nejdelší zdrojový kód: **254 řádků** (kapitola 20.2)

Textová podoba: obsah zaujme přibližně **90 stran čistého neformátovaného textu.**

ZÁVĚR

Hlavní myšlenkou této práce bylo vytvoření studijních podkladů k programu GNU Octave ve dvou variantách – jedna ve formě klasické referenční příručky a druhá jako HTML verze této příručky.

Tento program má pro vzdělávací instituce relativně slibnou budoucnost, jelikož je volně dostupnou variantou pro komerční software Matlab od firmy MathWorks. Oba tyto nástroje v sobě skrývají velký výpočetní potenciál a dají se použít ke složitým výpočtům, experimentům a simulacím, proto se pro školní použití zcela jistě hodí. Je však otázkou, jestli investovat do drahého softwaru, který poskytuje záruku kvality, stability, servisu a také svou rozšířeností zajišťuje široký okruh lidí – odborníků, kteří mohou být nápomocní.

Octave se Matlabu snaží přiblížit jak to jen jde. Ten, kdo umí pracovat s Matlabem by neměl mít problém přejít na Octave a opačně. GNUplot se taky vydařil a tvorba grafů probíhá téměř ve stejném duchu jako v konkurenčním programu – až na pár malých rozdílů. Pro Octave se na internetu v Octave-forge také vyskytují aplikační knihovny - toolboxy (opět analogie k Matlabu), které funkčnost programu značně rozšiřují - zpracování signálů, obrazu, návrh regulátorů, statistika, ...

Má snaha byla podat jak kvalitativní, tak kvantitativní výsledky ve formě volně dostupných podkladů ke studiu tohoto programu a principu jeho fungování. Více času jsem věnoval referenční příručce ve formátu HTML, protože tato technologie přenosu informací má mnohem větší potenciál vzhledem k dostupnosti internetu. WWW stránka také nabízí uživatelsky příjemnější prostředí také díky moderněji laděnému designu, možnosti odkazování a v neposlední řadě jsou stránky přehlednější.

Cílem byla co největší obsahová správnost překladu, avšak narážel jsem často na strohost originálního manuálu nebo naopak na přílišnou hloubku jednotlivých problematik. Proto jsem se snažil nalézt neoptimálnější cestu a studijní podklady přizpůsobit tak, aby bylo možné po jejich projití (či za jejich asistence) začít program plnohodnotně využívat.

SEZNAM POUŽITÉ LITERATURY

- [1] Wikipedie, otevřená encyklopedie. Dostupné z WWW:
<http://cs.wikipedia.org> [11.06.2006]
- [2] Dynamické HTML a kaskádové styly. Dostupné z WWW:
<http://www.kosek.cz/clanky/dhtml/styly.html> [11.06.2006]
- [3] XHTML – Kompletní průvodce. Dostupné z WWW:
<http://interval.cz/serialy/xhtml-kompletni-pruvodce/> [11.06.2006]
- [4] Maturita.cz – referáty kategorie Informatika - Internet. Dostupné z WWW:
http://www.maturita.cz/referaty/prehled_referatu.asp?kategorie=7 [11.06.2006]

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CSS	Cascading Style Sheets
DTD	Document Type Definition
ECMA	European Computer Manufacturers Association
FTP	File Transfer Protocol
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
ISO	International Standards Organization
ISOC	Internet Society
MathML	Mathematical Markup Language
PHP	Personal Home Page
RDT	Resource Description Framework
SGML	Standard Generalized Markup Language
SMIL	Synchronized Multimedia Integration Language
SMTP	Simple Mail Transfer Protocol
SVG	Scalable Vector Graphics
URL	Uniform Resource Locator
UTF-8	UCS Transformation Format
WWW	World Wide Web
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

SEZNAM OBRÁZKŮ

<i>Obr. 1</i> Reprezentace zdrojového kódu.....	17
<i>Obr. 2</i> Reprezentace absolutního umístění.....	22
<i>Obr. 3</i> Reprezentace relativního umístění.....	23
<i>Obr. 4</i> Okno s vlastním programem.....	35
<i>Obr. 5</i> Výpis matice.....	37
<i>Obr. 6</i> Omezený výpis matice.....	37
<i>Obr. 7</i> Výpis proměnné.....	38
<i>Obr. 8</i> Výpis násobení matice skalární hodnotou.....	38
<i>Obr. 9</i> Vykreslený graf funkcí sinus a cosinus.....	40
<i>Obr. 10</i> Vykreslený 3D graf.....	41
<i>Obr. 11</i> Komplikovanější 3D vykreslený objekt.....	41
<i>Obr. 12</i> Výpis nápovědy programu.....	42
<i>Obr. 13</i> Informace o programu TSW Webcoder.....	44
<i>Obr. 14</i> Efektivita: Zdrojová stránka.....	45
<i>Obr. 15</i> Efektivita: MS Word.....	46
<i>Obr. 16</i> Efektivita: Vlastnoručně psaná stránka.....	47
<i>Obr. 17</i> Pozadí stránky.....	48
<i>Obr. 18</i> Rozvržení tabulky.....	49
<i>Obr. 19</i> Finálně upravená tabulka a vzhled stránky.....	51
<i>Obr. 20</i> Ukázkový soubor při své reprezentaci v prohlížeči.....	52

SEZNAM PŘÍLOH

- P I OCTAVE - Referenční příručka základních příkazů
- P II CD-ROM (obsahuje samotnou bakalářskou práci, referenční příručku základních příkazů a její HTML verzi)