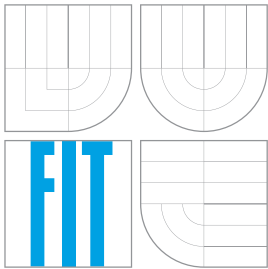


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## DOMÁCÍ MULTIMEDIÁLNÍ CENTRUM

HOME MULTIMEDIA CENTRE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DAVID SULAIMAN

VEDOUCÍ PRÁCE

SUPERVISOR

doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2008

## Abstrakt

Tato práce se zabývá návrhem a implementací multimediálního centra pro GNU/Linux. Práce začíná teoretickým úvodem, ve kterém je uveden přehled již existujících aplikací, operačních systémů, programovacích jazyků a současného hardwaru. Součástí jsou také výsledky ankety na téma multimédia v domácnosti. Ve druhé části najdeme kompletní návrh jádra systému, rozvržení modulů, nástin možnosti ovládání a krátké shrnutí testování během vývoje. V závěru jsou zachyceny výsledky práce a nástin dalšího vývoje.

## Klíčová slova

linux, java, mutlimédia, video, fotografie, hudba, přehrávač, vývoj

## Abstract

This beachelor's thesis is engaged in presentation of implementation of multimedia centre for GNU/Linux. The thesis begins with preambule summarized existing applications, operating systems, programming languages and current hardware. Another part analyses results of questionnaire on the theme household multimedia. In the second part there is full presentation of kernel of the system, scheme of modules, vignette of operating system and short summary of testing procedure beeing in procedure of program creation. A brief summary judges results of the work and proposal of the next system development.

## Keywords

linux, java, mutlimedia, video, photos, music, player, development

## Citace

David Sulaiman: Domáci multimediální centrum, bakalářská práce, Brno, FIT VUT v Brně, 2008

# Domáci multimediální centrum

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. RNDr. Pavla Smrže Ph.D.

.....  
David Sulaiman  
12. května 2008

© David Sulaiman, 2008.

*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

<b>1 Úvod</b>	<b>2</b>
<b>I Teoretický úvod</b>	<b>4</b>
<b>2 Přehled softwaru multimediálních center</b>	<b>5</b>
<b>3 Přehled dostupných technologií</b>	<b>7</b>
3.1 Operační systémy . . . . .	7
3.2 Programovací jazyky . . . . .	7
3.3 Hardware . . . . .	8
<b>4 Výsledky ankety</b>	<b>11</b>
<b>II Realizace systému</b>	<b>12</b>
<b>5 Jádru systému</b>	<b>13</b>
5.1 Zaslání zpráv . . . . .	13
5.2 Konfigurace . . . . .	14
<b>6 Moduly</b>	<b>17</b>
6.1 Výstupní modul . . . . .	17
6.2 Úvodní obrazovka . . . . .	17
6.3 Videopřehrávač . . . . .	18
6.4 Prohlížeč fotografií . . . . .	19
6.5 Hudební přehrávač . . . . .	20
<b>7 Ovládání</b>	<b>21</b>
7.1 Klávesnice . . . . .	21
<b>8 Testování</b>	<b>22</b>
<b>9 Závěr</b>	<b>23</b>
9.1 Výsledky . . . . .	23
9.2 Další vývoj . . . . .	23
<b>A Přílohy</b>	<b>27</b>

# Kapitola 1

## Úvod

Programy z oblasti multimediálních center nahrazují větší množství zařízení pro práci s multimédií, jako jsou DVD přehrávače a rekordéry, video rekordéry, HiFi věže a audiosystémy, umí obsluhovat televizní kartu a obsahují rozšiřující funkce. Mezi rozšiřující funkce patří například nástroje pro jednoduchý střih videa nebo jednoduché funkce pro práci s fotografiemi včetně jejich prohlížení. Tyto funkce se pokoušejí implementovat také výrobci DVD přehrávačů a rekordérů. Jsou to ovšem pouze jednoúčelová zařízení. U multimediálních programů se předpokládá jako výstupní zařízení televizor nebo větší monitor a ovládání pomocí dálkového ovládání. Hlavním rysem je centralizovanost, to znamená, že všechny funkce umožňuje jedna aplikace nikoliv několik specializovaných programů.

Každý běžný operační systém nabízí možnost přehrávání videa, hudby nebo prohlížení obrázků. Některé aplikace se pokouší integrovat více funkcí do sebe například Windows MediaPlayer nebo WinAmp. Není výjimkou, že tyto programy začínají být obtížně ovladatelné a uživatel se v nich hůře orientuje. Jejich funkcionalita není vždy zcela dotažena do konce. Například Windows MediaPlayer neumožňuje přehrávání DVD nebo vkládání titulků k filmům a je nutno použít externí program, který titulky vkládá. Ovládání takovýchto programů je koncipováno pro práci s myší ev. klávesnicí, nikoliv např. pomocí dálkového ovládání. Řešení této situace nabízejí multimediální centra, která se snaží přizpůsobit svou funkcionalitu požadavkům uživatelů a jsou koncipována pro snadné ovládání myší, klávesnicí ale i dálkovým ovladačem.

Multimediální aplikace jsou buď samostatné aplikace spuštěné v určitém operačním systému nebo mohou být se systémem provázány a spouští se automaticky po startu systému. Běžný uživatel ani nezaznamená start operačního systému, který je vhodně skryt. Výhodou integrovaných aplikací je stoprocentní přizpůsobení systému pouze pro potřeby aplikace a vyznačuje se rychlejším startem a nižšími nároky na výkon počítače. Oproti tomu řešení s využitím standardního operačního systému nabízí možnost využít běžných programů bez restartu systému nebo zároveň pracovat společně při přehrávání médií například na druhé obrazovce. Programy, které existují v integrovaném provedení, se také vyskytují v provedení pro běžný operační systém.

Na poli operačních systémů jsou majoritní systémy společnosti Microsoft a pomalu sílící GNU/Linux. Protože jsem zastánce a dlouholetý uživatel a administrátor systému GNU/Linux, rozhodl jsem se vytvořit aplikaci multimediálního centra pro GNU/Linux. Bude snadno rozšiřitelné a upravitelné, na což jsou právě uživatelé GNU/Linuxu zvyklí.

Jednou z možností řešení, které by odpovídalo linuxovému principu, je vytvořit pouze uživatelské rozhraní a využívat již hotové aplikace pro přehrávání. Tato varianta je však silně omezující a je vázána na konkrétní aplikace. Také možnosti úprav a rozšíření stávajících

přehrávačů jsou mizivé nemluvě o rozdílném ovládní. Zvolil jsme nejflexibilnější variantu, a to program složený z jádra a modulů/pluginů, které se budou start o vlastní přehrávání, zobrazování nebo ovládní.

První část této práce je teoretický úvod (I), který obsahuje nástin již vytvořených programů pro multimediální centra (2). Obsahuje také přehled rozdílů mezi operačními systémy Microsoft Windows, MacOS a GNU/Linux, shrnutí programovacích jazyků dostupných v Linuxu. Kapitola (3) se zabývá současným hardwarem z pohledu na multimediální využití. Závěr této části obsahuje stručné shrnutí ankety o multimédiích v domácnosti (4), kterou jsem v rámci práce provedl.

Druhá část popisuje realizaci systému (II) od návrhu jádra systému (5), přes návrh a realizaci modulů (6). Dále jsem se zabýval ovládním systému (7) a závěrečná kapitola se věnuje testování (8).

Část I

**Teoretický úvod**

## Kapitola 2

# Přehled softwaru multimediálních center

Software implementující multimediální centra již nějakou dobu existuje, stále se vyvíjí a zároveň vznikají i varianty, které se plnohodnotnému centru jen podobají. Software pro multimediální centrum lze zakoupit nebo získat společně s jiným softwarem, například Nero<sup>1</sup>, software také dodávají výrobci hardwaru, například výrobci televizních karet. Drtivá většina těchto karet dodává programy pouze pro operační systém Microsoft Windows, tyto programy nejsou příliš propracované. Lepší varianty jsou Windows Media Center [6] nebo software vyvinuté speciálně pro práci s prodávaným zařízením. Jako příklad bych uvedl špičkový software od společnosti Pinnacle nebo již ne tak zdařilý od společnosti AverMedia.



Obrázek 2.1: Windows XP Media Center Edition 2005, zdroj [16]

<sup>1</sup>Software pro vypalování, dodáván k většině CD/DVD vypalovaček



Podpora GNU/Linuxu výrobci je jen nepatrná, ve většině případů nedodávají ovladače ani software. Ten ovšem není zcela podstatný. Linuxová komunita tyto problémy řeší tvorbou vlastních ovladačů, které integrují přímo do jádra systému. Také open source software, který vzniká, je mnohdy kvalitnější než software dodávaný. Je třeba říci, že ten v některých případech slouží pouze pro ukázkou funkčnosti zařízení. V každém případě je věnováno více nákladů na výrobu hardwaru než doprovodného softwaru. Jedinou výjimku v tomto ohledu zřejmě tvoří společnost Pinnacle.

Jak jsem se již zmínil, Linux řeší software po svém. Je vyvíjeno několik multimediálních center, například Linux MCE [3] označovaný jako all-in-one, který zahrnuje obsahuje vše nač si jen vzpomenete. U Linuxu není takový rozdíl mezi programem spouštěným v klasické linuxové distribuci nebo upravenou distribucí. Linux sám o sobě je velice variabilní a přizpůsobivý. Existují však již hotová řešení multimediálního centra bez nutnosti další instalace a administrace systému nebo tak zvané Live distribuce, které se spouštějí z CD a po načtení systému do paměti umožní toto CD vyměnit za multimediální titul.



Obrázek 2.2: Linux MCE, zdroj [3]

Dalším velice zajímavým centrem, bohužel bez možnosti práce s televizí, je Elisa [1]. Na první pohled vypadá velice zdařile, nepodařilo se mi ovšem po instalaci spustit film a ani lokální soubor s hudbou. Varantou řešení pomocí Live CD je GeeXboX [2], který je postaven na programu MythTV [7]. Za zmínku stojí, že celá distribuce o velikosti pouhých 6,8MB se zkopíruje do paměti a tím umožňuje vyjmutí zdrojového CD a vložení požadového média.

## Kapitola 3

# Přehled dostupných technologií

### 3.1 Operační systémy

Na poli operačních systémů válčí tři druhy. Mac OS X, který se standardně dodává s počítači Macintosh firmy Apple. Jeho velkou předností je velice ergonomický způsob práce a vysoká úroveň uživatelského rozhraní. Tento systém má pokročilé technologie pro práci s multimédií. Je možné ho nainstalovat i na počítače typu PC, ale i když patří mezi silně minoritní systémy, má své uživatele.

Na většině prodávaných PC nebo notebooků je již výrobcem nebo prodejcem instalován operační systém Microsoft Windows. Tato situace se však pomalu mění a přicházejí osobní počítače s GNU/Linux. Microsoft patří mezi největší dodavatele operačních systémů pro PC již od jejich počátku. První operační systém, který byl dodán toto společností pro společnost IBM byl 16bitový MS DOS. Následovaly grafické nadstavby, po vzoru programů společnosti Apple, Windows 3 a Windows 3.11. Největším skokem bylo uvedení Windows 95, které již nebyly pouze grafickou nadstavbou DOSu a měli nativní grafické rozhraní. Následovala řada systémů Windows 98, NT, Milenium, 2000. Dnešnímu trhu dominují Windows XP a Windows Vista. Každá nová verze přináší změny, některé jsou více pozitivní některé méně.

Posledním systémem, který postupně zvyšuje svůj podíl na trhu je GNU/Linux. Autorem jádra systému Linux je Linus Torvalds. Tento systém je tvořen jádrem a skupinou programů, kterou má možnost si každý uživatel upravit. Je vydáván v mnoha distribucích, které se od sebe liší sadou standardně dodávaných programů, správou konfigurace systému, balíčkovacím systémem<sup>1</sup> a v neposlední řadě přístupem k uživateli. Linux ve svém počátku byl textový stejně jako Microsoft DOS, ale má mnoho grafických nadstavb. Linuxová koncepce již od počátku vychází s multiuživatelského a multiprocesového prostředí. Toto s sebou nese výhody i nevýhody. Pro běžného uživatele to může být i vhodně skryto, avšak ponechává si své flexibilní a bezpečnostní aspekty. Z hlediska multimediálního použití je již rovnocenný s Microsoft Windows, u kterých to bylo vždy lákadlem.

### 3.2 Programovací jazyky

Jelikož jsem zvolil jako prioritní operační systému GNU/Linux, vyplývají z toho i použitelné programovací jazyky. Pro GNU/Linux je přeložitelná a spustitelná velká škála jazyků,

<sup>1</sup>V Linuxu se instalovatelný program nazývá balíček. Existují nástroje pro snadnou instalaci, správu a odebrání balíčku.

včetně například jazyka Pascal, který je určen hlavně pro studijní účely, nebo též Assembler, jazyk symbolických instrukcí, ve kterém by byla práce na takto rozsáhlém projektu jen velice těžko provídatelná. Dalším kritériem byla i značná rozšířenost jazyka, aby moduly mohli vytvářet i programátorsky schopní uživatelé systému.

Jazyk, který k Linuxu neodmyslitelně patří, je jazyk C. Je to jazyk umožňující strukturované programování, přímý přístup do paměti pomocí ukazatelů a tvoří velmi kompaktní aplikace. Je v něm napsáno například jádro operačního systému Linux, překladač gcc a další. Pro nás není od věci, že přehrávač MPlayer je také napsán v C.

C++ je objektově orientovaný jazyk vytvořený jako rozšíření jazyka C. Jazyk C je tudíž podmnožinou jazyka C++ ovšem až na několik přesně definovaných výjimek. Oba jazyky je možno přeložit překladačem pro C++. C++ však není čistě objektově orientovaný jazyk a podporuje několik programovacích stylů (paradigmat), procedurální, objektově orientované a generické programování.

Perl je reprezentant interpretovaného programovacího jazyka. V začátcích rozvoje internetu se používat pro tvorbu CGI skriptů. CGI je protokol pro propojení webového serveru a externí aplikace. Jazyk byl vytvořen jako náhrada jazyka AWK a interpretu sh. Motem toho jazyka je heslo: “dá se to udělat více způsoby”. Umožňuje psát složité programy s vhodnou strukturou, ale dovolí napsat i krátký program, a to velice rychle. Programátor bez větších zkušeností však snadno vytvoří nesrozumitelný a špatně udržovatelný kód.

Python je stejně jako předchozí Perl jazyk interpretovaný. Oba patří mezi takzvané skriptovací jazyky. Python však má větší možnosti a byl navržen tak, aby umožňoval tvorbu velkých aplikací obsahující grafické rozhraní, které se může tvořit v různých variantách (wxPython, PyQt..). Python patří mezi nejsnáze osvojitelné programovací jazyky a je také důkazem, že jazyk vhodný pro výuku je také vhodný k praktickému použití. Stejně jako Perl umožňuje psát krátké a rychlé programy. Na druhou stranu obsahuje rysy pro psaní velkých projektů, například podporu prostorů jmen, zachytávání výjimek a prostředky pro tvorbu testů.

Společnost Sun Microsystems v roce 1995 představila objektový jazyk Java[17]. Patří mezi nejpoužívanější jazyky na světě, a to díky své přenositelnosti. Může být spuštěna na mnoha platformách jako jsou mobilní telefony a zabudovaná zařízení, desktopové počítače s různými operačními systémy i distribuované systémy. Je to jazyk interpretovaný, avšak místo skutečného strojového kódu se vytváří mezikód (bytecode), který je pak interpretován virtuálním strojem na libovolném zařízení. Je-li nějaká část kódu často používaná, interpret provede optimalizaci a bytecode přeloží do strojového kódu. Java neumožňuje některé programátorské konstrukce, tímto přispívá k tvorbě spolehlivého softwaru. Zprávu paměti zajišťuje takzvaný Garbage collector, který vyhledává nepoužívané části paměti a uvolňuje je pro další použití. Nyní je Java vyvíjena jako open source.[14]

Z těchto jazyků jsem si zvolil Javu, která přináší výhody přenositelnosti a je rozšířena mezi programátory. Další výhodou je snadná tvorba uživatelského rozhraní pomocí třídy Swing, která se nachází v javax.swing. Práce Javy s multimédií není ovšem příliš rozvinutá. Existuje balík Java Media Framework (JMF), který je jedinou možností přehrávání a práce s videem. [19]

### 3.3 Hardware

Osobní počítače, se kterými se setkáváme v dnešní době jsou již tak výkonné, že nemá smysl uvažovat na omezením při přehrávání multimédií. Samotné operační systémy společnosti

Microsoft kontrolují při instalaci minimální požadavky systému. Například moderní Windows Vista požaduje minimálně 512MB paměti. Tato velikost je již dostačující ke spuštění téměř libovolných aplikací včetně programů psaných v Javě.

Pro přehrávání hudby jsou k dispozici různé varianty zvukových zařízení. Jako standard se dnes bere 16 bitová šířka pásma<sup>2</sup>, u kvalitnějších zvukových karet dosahuje 24bitů. Výstup zvukových karet může být buď analogový vyvedený pomocí konektorů jack nebo digitální SPDIF, který může být přenášen pomocí vodičů nebo optických kabelů.

Z pohledu mé práce je nejdůležitějším kritériem počet výstupních kanálů. Tato dvojice čísel ve formátu X.Y udává počet malých reproduktorů X rozmístěných po místnosti, tak zvaných satelitů. Nejčastěji v počtech 2,4,5 a 7. Druhé číslo Y udává počet basových reproduktorů, takzvaných subwooferů. Nejčastějším provedením je 5.1, kde jsou 4 satelity rozmístěny ideálně v rozích místnosti, středový satelit nad obrazovkou. Na pozici subwooferu příliš nezáleží.

Při přehrávání filmů je nutné zajistit odpovídající zobrazení obrazu. Na tom se podílejí dvě komponenty, grafická karta a monitor popřípadě televizor. U současných grafických karet se již neřeší problém se zobrazováním určitého rozlišení a vysokého počtu barev, takzvané barevné hloubky. Doménou dnešních karet je zobrazování trojrozměrných scén a zobrazování libovolného videa jim nečiní problém.

Původní CRT<sup>3</sup> monitory, které byly připojovány pomocí analogového konektoru VGA (Video Graphics Array), se z relativně malých 13" černobílých monitorů se postupně vyvinuly v barevné monitory dosahující velikosti až 22" a plně barevného obrazu. Mezi CRT monitory se postupně začaly prosazovat LCD<sup>4</sup> displeje, které nabízejí větší plochu až 24" a výrazně tenčí provedení však s nižším rozlišením. Možnost připojení pomocí VGA a zároveň digitálního rozhraní DVI (Digital Visual Interface) je výhodou. Rozhraní DVI existuje ve dvou variantách DVI-D, které obsahuje pouze digitální signál a DVI-I, který obsahuje oba druhy signálu, digitální i analogový. Z rozhraní DVI-I je možné extrahovat pouze analogový signál a získat tak rozhraní VGA. Mezi novinky patří rozhraní HDMI (High-Definition Multimedia Interface). Toto rozhraní je čistě digitální s možností přenášet video v kvalitě high-definition a zvuku až o osmi kanálech. HDMI je možno také získat z rozhraní DVI-D nebo DVI-I ovšem bez možnosti přenosu zvuku. Ten se musí přenášet zvlášť.

K připojení klasického televizoru se využívá konektoru a normy S-Video, jehož výstup však musí být přítomný na grafické kartě. Tato norma přenáší signál analogově v rozlišení klasického televizoru. Nemá-li televizor konektor pro S-Video existují redukce mezi S-Video a cinch a také S-Video a scart. LCD televizory již disponují vstupy pro připojení PC. Mezi standardní rozhraní patří HDMI, které se využívá i k připojení ostatních zařízení, dále bývá k dispozici VGA či DVI.

K přehrávání běžných multimediálních disků DVD a CD jsou počítače vybaveny vhodnými mechanikami, které umožňují nejen čtení, ale také zápis. Zdrojem multimédií nemusí být jen CD nebo DVD ale také soubory uložené na lokálních pevných discích, jejichž kapacity již překročili 500GB. Dalším zdrojem může být internet. Internetová rádia jsou již běžná a například Česká televize má na internetu dostupné své pořady.

Ovšem mezi nejstarší média patří rozhlasové a televizní vysílání. Aby bylo možné přijímat tyto technologie, je nutné mít televizní kartu. Existují tři typy provedení, analogová, digitální a hybridní, která kombinuje předchozí dvě možnosti. Karta analogová slouží

<sup>2</sup>Počet bitů převedených na audiosignál

<sup>3</sup>Cathode ray tube - katodová trubice

<sup>4</sup>Liquid crystal display - display z tekutých krystalů

k příjmu analogového televizního a radiového vysílání, které je dostupné na většině našeho území. Na zbytku, kde již proběhla tak zvaná digitalizace, se využije karta digitální. Ovšem nákup karty analogové je silně neperspektivní záležitost, vznikla tedy proto karta hybridní umožňující přijímat oba druhy vysílání.

K těmto kartám se ve velké většině případů dodává dálkové ovládání. Bohužel většina výrobců je dodává jen malá a ne příliš ergonomicky řešená. Pomineme-li náhradu dodaného dálkového ovladače univerzálním ovladačem, který ve vyšší cenové kategorii bývá značně propracovaný včetně dotykového displeje, kompletní programovatelnosti a tvorby maker, nabízí se možnost koupě samostatného ovládání.

Dálkové ovládání je možno realizovat s využitím infračerveného signálu, jako je to například u televizních přijímačů nebo pomocí radiového signálu. Výhodou radiového řešení není nutnost přímé viditelnosti mezi ovladačem a přijímačem. Jinak je tomu u infračervených ovladačů. Ty jsou ovšem častější než radiové a vyrábí se například s přijímačem do 5,25" pozice<sup>5</sup>, který obsahuje také display pro zobrazení informací právě pro multimediální centra. Společnost vyrábí se také počítačové skříně obsahující infračervený přijímač a jsou vzhledově podobné ostatní elektronice.

---

<sup>5</sup>Pozice na přední straně počítačové skříně, umísťují se tam například CD/DVD mechaniky

## Kapitola 4

# Výsledky ankety

Pro zjištění využitelnosti možných funkcí multimediálního centra jsem provedl výzkum formou ankety, které se zúčastnilo 30 respondentů, ve složení 43% muži a 57% ženy. Věk respondentů se pohyboval v rozmezí od 15 do 60 let. Nejvíce dotazovaných bylo ve věku od 15 do 35 let (80%). V 93% vlastní osobní počítač nebo notebook a všichni mají přístup k internetu.

Počítač k přehrávání filmů používá 50% respondentů a z toho 20% dotazovaných osob má propojený počítač s klasickou nebo LCD televizí. 23% z celkového počtu účastníků ankety používá k přehrávání filmů počítač pouze vyjíměčně.

Následující tabulka shrnuje rozložení sestav reproduktorů u dotazovaných osob:

Rozložení reproduktorů	Počet procent
2.0	74%
2.1	7%
5.1	15%
6.1	4%

Není trendem využívat počítač jako televizní přijímač, většina lidí využívá klasickou televizi a tomu odpovídal i výsledek. 75% dotazovaných nevlastní televizní tuner. Majitelé analogového a digitálního tuneru jsou zastoupeni shodně 11%. Hybridní tuner vlastní pouze 4% respondentů.

Z výsledku ankety vyplývá, že aplikace se obejde bez možnosti nahrávání vysílaných pořadů, protože celá polovina dotázaných pořady nenahrává a těch, kteří nahrávají je pouze 13%. Zbytek nahrává pouze minimálně nebo vyjíměčně.

Nejvíce využívaným formátem videa je DVD, následuje televizní vysílání. Komprimované formáty DivX a MPEG obsadili třetí a čtvrtou pozici v žebříčku oblíbenosti a tím překonali formát HD video, který se neočekávaně umístil až na posledním místě. Zřejmým důvodem je jeho malá rozšířenost v prodeji a pro plné využití je nutno vlastnit celkem nákladnou plazmovou nebo LCD televizi podporující HD rozlišení.

63% respondentů poslouchá živé rádiové vysílání a 40% dotázaných se věnuje také poslechu digitálních skladeb ve formátu audio CD a MP3. Mezi nejoblíbenější nosiče videa patří disky DVD, následují pevné disky společně s CD. Pouze 17% respondentů nepoužívá výpočetní techniku k prohlížení fotografií, a proto se multimediální aplikace obejde bez prohlížeče fotografií.

Část II

**Realizace systému**

## Kapitola 5

# Jádro systému

U aplikace, která má být snadno rozšiřitelná, upravitelná či skinovatelná<sup>1</sup>, je nejdůležitější součástí jádro. Tato část programu zajišťuje nahrávání modulů, jejich správu a spouštění, musí umožňovat práci s konfiguračním souborem a také umožnit modulům mezi sebou komunikovat.

V programovacím jazyku Java nebyl problém implementovat načítání modulů, až když bude vyžadováno jejich spuštění. Pro správu modulů je vyhrazena třída `ModuleManager`, která obsahuje metody pro nahrávání tříd a také metody pro vyhledávání v již nahraných modulech.

Jako modul je označena část programu, která zajišťuje jednu činnost, například přehrávač videa. Všechny moduly mají svou nadřazenou třídu, která je umístěna v balíku `pluginClasses`. Tyto třídy zajišťují základní inicializaci nahrávaného modulu, mimo jiné i odstranění povolené zprávy `Init` z přijímaných zpráv. Další funkcí nadřazené třídy je definování povinných akcí, které by měly mít v modulu svou funkcionalitu. Pro názornost uvádím další příklad: modul pro zobrazení úvodní obrazovky musí reagovat vždy na zprávy o pohybu, proto má jeho nadřazená třída `InitScreen` nadefinovány abstraktní metody pro tyto zprávy.

Definoval jsem tyto nadřazené třídy:

- `Controller` - třída ovládacích zařízení: klávesnice, myš
- `InitScreen` - třída pro úvodní obrazovky
- `Output` - třída pro zobrazování výstupu
- `Player` - třída pro moduly přehrávačů
- `Module` - nadtřída zde uvedených tříd, obsahuje metody pro práci se zprávami

### 5.1 Zasílání zpráv

V modulovém systému je nutné mít definovanou komunikaci mezi jednotlivými částmi systému. Aby bylo rozhraní dostatečně univerzální, je vytvořena třída `Message`, která obsahuje název zasílané zprávy a další doplňující data. Ta jsou zatím omezena pouze na `String` a `Component`. V tuto chvíli nebylo potřeba více typů. Zpráva se dá adresovat libovolnému

---

<sup>1</sup>změna vzhledu aplikace



modulu nebo jejich nadtrídě. Pokud jako adresát bude uvedena třída `Module` bude zpráva doručena všem modulům.

Zpráva se zasílá metodou `messageSender(Message message, Class<?> class)`. Tato metoda využije `ModuleManager` k získání všech načtených modulů, které jsou odpovídají třídě `class`. Následně je u každého adresovaného modulu volána metoda `messageReceiver`. Ta získává z `actionMap` podle textu zprávy příslušnou metodu, kterou spustí a jako jediný parametr jí předá zprávu `Message`.

Chce-li modul přijímat zprávy s určitým textem, musí je mít zaregistrované, k tomu slouží metoda `addInActionList(String actionText, Method method)` implementovaná v nejvyšší nadtrídě `Module`. Registrace zprávy spočívá v uložení klíče `actionText` do hash mapy `actionMap` a jako hodnota je uložena přiřazená metoda `method`. Objekt `actionMap` je reprezentován pomocí `HashMap` pro rychlé vyhledávání podle klíčů.

Pro snadnější využívání byla metoda `addInActionList(String actionText, Method method)` přetížena metodou `addInActionList(Message actionMessage, Method method)`. Výhodou je jednodušší zápis, který programátora nenutí ze zprávy extrahovat její text, ale může předat celou zprávu. V metodě je ze zprávy získán její text a pak je volána metoda `addInActionList` používající pouze text zprávy a jemu přiřazenou metodu.

Objekt do třídy `Message` byl přidán dodatečně z nutnosti nepředávat zprávou pouze příkazy, ale i objekty, které se mají využít. Například při zobrazování prvků jsou volány zprávy pro moduly nadtrídě `Output` a jako `Component` mají objekt pro vložení na zobrazovací plochu. Ve třídě zprávy jsou pouze dva datové objekty, ale třída může být rozšířena. Konkrétní objekty na místo objektu `Object` byly zvoleny jako bezpečnostní opatření, aby nedošlo omylem ke špatnému přetypování.

## 5.2 Konfigurace

Aplikace tohoto rozsahu již potřebují možnosti konfigurace. Vhodné a zároveň čitelné uložení konfigurace je ve formátu XML (Extensible Markup Language) což v překladu znamená rozšiřitelný značkovací jazyk. Jazyk XML byl vyvinut pro snadné vytváření značkovacích jazyků. Jeho určení je pro přenos dat mezi aplikacemi nebo pro ukládání strukturovaných dat bez důrazu na vzhled.[13]

Výhodou XML je snadná čitelnost člověkem i počítačem.

Pro práci s konfiguračním souborem byla vytvořena třída `Config`, jejíž návrh jsem značně podcenil. Při návrhu byla diskutována pouze forma, která využila návrhového vzoru singleton<sup>2</sup>. Ovšem první slabina, která se projevila až při implementaci byl jediný konfigurační soubor `config.xml`. Ten se sice při inicializaci třídy pro konfiguraci načte, ale zabránil přidružení konfigurace k jednotlivým modulům. Dalším přetrvávajícím problémem je špatně navržené API<sup>3</sup>.

Správněji řečeno API nebylo navrženo vůbec. Tato třída pouze načte jeden konfigurační soubor a na základě několika metod předává data z XML. Metody ovšem nejsou parametrizované a tím pádem jsou jednoúčelné. Pro práci s XML soubory jsem využil knihovnu `dom4j`[9].

---

<sup>2</sup>Cílem vzoru je zajištění existence pouze jedné instance dané třídy a poskytnutí globálního přístupu k ní.[18]

<sup>3</sup>API je zkratka anglických slov application programming interface, což znamená rozhraní pro programování aplikací. Tento termín používá softwarové inženýrství v programování. Jde o sbírku procedur, funkcí či tříd nějaké knihovny (ale třeba i jiného programu nebo jádra operačního systému), které může využívat programátor, který knihovnu využívá.[13]

Vhodnějším návrhem konfigurovatelnosti systému je vytvoření více XML souborů pro každý modul a pro jádro systému. Také metody ve třídě `Config` by měli být parametrizovatelné, aby dokázaly požadovaná data předávat příslušným modulům.

Jak již bylo zmíněno konfigurační data jsou ukládána ve formátu XML v souboru `config.xml`. Použil jsem univerzální kódování UTF-8, což je uvedeno v hlavičce souboru.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Správný XML soubor má obsahovat jeden hlavní element, ve kterém jsou obsaženy další informace. Hlavním elementem konfigurace je element, který je otevřený značkou `<config>` a uzavřený `</config>`. V konfiguraci je uveden výčet nahrávaných modulů/pluginů pro výstup na obrazovku `<outputPlugin>`, úvodní obrazovku `<initPlugin>` a také ovládací prvky `<controllerPlugin>`. Konfigurační element, který informuje systém o dostupných přehrávačích a také ho využívá úvodní obrazovka, je element `<players>`. Všechny tyto elementy obsahují elementy s názvem `plugin`, které mají povinný parametr `name`.

```
<plugin name="plugins.InitScreen.DefaultInitScreen" />
```

Element `plugin` pro použití v elementu `<players>` je ještě rozšířen o druh přehrávaného média `class` a parametr zdroje dat `medium`.

```
<plugin class="picture" name="plugins.PictureExplorer" medium="dir" />
```

Dále v souboru následuje konfigurace pro modul/plugin `InitScreen`, což je modul úvodní obrazovky. Obsahuje pouze seznam obrázků pro zobrazení tlačítek. Tento seznam je vložen v elementu se stejným názvem jako je název modulu. Obsahuje elementy `picture` s parametry `usage`, `nonFocus` a `focus`.

Parametr `usage` je textový popis použití obrázků. Je-li uveden pouze formát média, například `video` bude zobrazen pro celou skupinu. Následuje-li pomlčka a typ zdrojového média, například `video-file`, jsou obrázky využity pro konkrétní tlačítko spouštějící daný modul. V tomto modulu je jediná výjimka, a to u `usage='exit'`, kde tento element zahrnuje obrázky pro tlačítko ukončení programu.

Parametry `nonFocus` a `focus` jsou cesty k obrázkům pro nevybraný (`nonFocus`) a vybraný prvek (`focus`). Názorná ukázka na příkladu.

```
<plugins.InitScreen.DefaultInitScreen>
  <picture usage="video"
    nonFocus="./src/plugins/InitScreen/film1.png"
    focus="./src/plugins/InitScreen/film.png" />
  <picture usage="video-file"
    nonFocus="./src/plugins/InitScreen/tlacitko1.png"
    focus="./src/plugins/InitScreen/tlacitko.png" />
  <picture usage="video-DVD"
    nonFocus="./src/plugins/InitScreen/tlacitko1.png"
    focus="./src/plugins/InitScreen/tlacitko.png" />
  <picture usage="picture"
    nonFocus="./src/plugins/InitScreen/picture1.png"
    focus="./src/plugins/InitScreen/picture.png" />
  <picture usage="picture-dir"
    nonFocus="./src/plugins/InitScreen/dir1.png"
    focus="./src/plugins/InitScreen/dir.png" />
```

```
<picture usage="exit"  
    nonFocus="./src/plugins/InitScreen/offbutton1.png"  
    focus="./src/plugins/InitScreen/offbutton.png" />  
</plugins.InitScreen.DefaultInitScreen>
```

# Kapitola 6

## Moduly

### 6.1 Výstupní modul

Výstupní modul `Output` slouží jako správce zobrazování obrazu. Mnou vytvořená implementace `SimpleOutput` zajišťuje zobrazení okna a jeho maximalizaci do celobrazového režimu. Umožňuje také skrývat kurzor myši a zajišťuje připojení modulů nadtržidy `Controller`. Zobrazovaný rámec, jehož layout je nastavený na `GridLayout()`, reprezentuje proměnná `frame`.

S ostatními moduly je provázán pomocí dvou zpráv, `addElement` a `clearOutput`. První zpráva vloží do zobrazovaného rámce komponentu, kterou zpráva obsahuje a je možné ji získat pomocí metody `getComponent()`. Metoda `clearOutput` reagující na stejnojmennou zprávu vymaže všechny komponenty, které byly předány společně se zprávou `addElement`. Tato možnost se využívá při přechodu mezi moduly nebo kompletní změně vzhledu. Jako například lze uvést přechod z úvodní obrazovky na výběr adresáře pro zobrazování fotografií.

Jaký modul bude použit pro zobrazování, získá systém z konfiguračního souboru pomocí metody `getOutputPlugin()` ve třídě `Config`. Zápis vyjadřující název modulu v konfiguračním souboru vypadá takto:

```
<outputPlugin>
  <plugin name="plugins.SimpleOutput" />
</outputPlugin>
```

Vytvořil jsem grafické rozhraní jako samostatnou aplikaci. Je však možné naprogramovat výstupní modul jako applet, komponentu běžící v kontextu jiného programu, typicky webového prohlížeče. Applet se spouští na straně klienta. Ovšem jeho použití je spíše pro vysokorychlostní nebo lokální síť

### 6.2 Úvodní obrazovka

Tento modul podtržidy `InitScreen` je nahrán systémem bezprostředně po spuštění systému a umožňuje uživateli navigaci v systému. Jeho název a umístění je uloženo v konfiguračním souboru ve formátu:

```
<initPlugin>
  <plugin name="plugins.InitScreen.DefaultInitScreen" />
</initPlugin>
```

Pro tvorbu grafického vzhledu bylo nutné vytvořit dvě nové třídy `PictureButton` a `PicturePanel`. Jedná se o rozšíření prvků `Button` a `Panel`.

Úpravy ve třídě `Panel` nebyly nijak výrazné. Doplnil jsem privátní proměnnou typu `BufferedImage` a k ní metodu `getImage()`, která vrací obrázek, a druhou metodu `setImage(BufferedImage img)`, která ukládá obraz do proměnné, ale také nastavuje preferovanou velikost komponenty na rozměry obrázku. Poslední změnou ve třídě bylo překrytí metody `paintComponent(Graphics g)`. Přidal jsem vykreslování obrázku na pozadí komponenty. Avšak změnila-li se velikost oproti původnímu obrázku, byla provedena standardně nepříliš kvalitní interpolace<sup>1</sup> a její důsledek byl patrný v podobě čtverečků. Bylo proto nutné použít kvalitnější, v mém případě bikubickou interpolaci, kterou obsahuje třída `Graphics2D`. Velmi dobrou vlastností je možnost předání `null` místo objektu obrázku `BufferedImage`. `Panel` se chová zcela běžně jak je očekáváno, ovšem jeho pozadí se nezobrazí a jeví se jako průhledné a toho se dá velice dobře využít.

Zdrojový kód metody `paintComponent(Graphics g)`:

```
public void paintComponent(Graphics g) {
    super.paintComponent(g);

    Graphics2D g2d = (Graphics2D)g;
    g2d.setRenderingHint(RenderingHints.KEY_INTERPOLATION ,
        RenderingHints.VALUE_INTERPOLATION_BICUBIC);
    if (image != null)
        g2d.drawImage(image,0,0,this.getWidth(),this.getHeight(),null);
}
```

Třída `PictureButton` je ještě poněkud složitější. Používá se zde stejný princip vykreslování jako u třídy `PicturePanel`. Je zde ovšem problém v tom, že tlačítko má více vzhledů a musí tedy obsahovat více obrázků, pro zaměřené a nezaměřené tlačítko. Změnu vzhledu obstarávají metody `setFocused()` a `setNonFocused()`, které proměnné reprezentující vykreslovaný obrázek, přiřazují ten vybraný. Také bylo nutno definovat reakce na myš, aby při stisku tlačítka došlo ke změně jeho vzhledu.

Další třídou vytvořenou pro modul úvodní obrazovky je `Focuser`. Tato třída nahrazuje klasické zaměrování prováděné pomocí klávesy tabulátoru. Podstatným rozdílem je možnost využívat k ovládní zprávy, které modul získává od rozhraní ovládní klávesnice. Druhou výhodou je přeskokování momentálně neviditelných tlačítek a tím přirozený pohyb jsou-li některá tlačítka skryta.

## 6.3 Videopřehrávač

Funkce, bez které by multimediální centrum nebylo multimediálním centrem, je přehrávání videa. To je zprostředkováno pomocí modulu videopřehrávače. Ten je implementací třídy pro přehrávání `Player`.

Pro přehrávání se v Javě používá knihovna JMF, která je součástí standardní instalace, avšak neumožňuje přehrávat komprimovaná média. Řešení nabízí rozšíření pro JMF jménem FOBS, které využívá knihovnu FFmpeg pro dekodování videa.

Při inicializaci třídy `Player` se zašle zpráva `clearOutput` modulům `Output`, která má za následek vymazání všech prvků z hlavní komponenty. Následuje načtení video souboru a tvorby přehrávače pomocí `Manager.createRealizedPlayer(DataSource source)`.

<sup>1</sup>proložení bodu křivkou a následné převedení na body ovšem v požadované velikosti

Po odeslání zprávy obsahující panel s černým pozadím se vytvoří vizuální komponenta metodou `getVisualComponent()`, která je vložena do černého panelu.

Pro zajištění správné změny velikosti stran u komponent jsem implementoval metodu `resizeKeepScale(Component component, Component panel)`, která podle velikosti panelu změní velikost komponenty tak, aby nepřesahovala omezující panel a zároveň zachovávala poměr stran. Tato metoda je umístěna v modulu `Module` a je dostupná pro všechny moduly.

```
protected void resizeKeepScale(Component component, Component panel){
    Dimension originalSize = component.getPreferredSize();

    double scaleWidth=
        (double)panel.getSize().width/(double)originalSize.width;
    double scaleHeight=
        (double)panel.getSize().height/(double)originalSize.height;

    double scale = 1;
    if (scaleHeight > 0 && scaleWidth > 0){
        if ( ((scaleWidth*originalSize.height) <= panel.getSize().height)){
            scale = scaleWidth;
        }else{
            scale = scaleHeight;
        }
    }else{
        if (scaleHeight > 0)
            scale = scaleHeight;
        if (scaleWidth > 0)
            scale = scaleWidth;
    }

    int width = (int)(originalSize.width * scale);
    int height = (int)(originalSize.height * scale);

    component.setPreferredSize(new Dimension(width, height));
    component.setSize(new Dimension(width, height));
}
```

## 6.4 Prohlížeč fotografií

Pro prohlížení fotografií nebo obrázků jsem udělal další implementaci třídy `Player`. Funkcionalita se skládá ze dvou kroků, výběr adresáře obsahující fotografie a jejich následné prohlížení. Jako prohlížené formáty jsem zvolil JPEG a PNG. Java umí více formátů, ale myslím, že pro prohlížení fotografií toto omezení vyhovuje. Stejně jako u přehrávače videa se fotografie zobrazují na černém pozadí a jsou zmenšovány nebo zvětšovány, aby maximálně využily plochu obrazovky.

Vybraný adresář nemusí přímo obsahovat fotografie, modul prochází všechny podadresáře a vytváří seznam všech fotografií, které jsou v nich obsaženy. Zde se uplatnil algoritmus procházení do hloubky. Nejdříve seřadí seznam souborů dané úrovně a následně vstoupí

do prvního adresáře. Tento postup se opakuje, dokud má adresář podadresáře, následuje procházení dalšího adresáře. Takový postup je příkladem rekurzivního<sup>2</sup> algoritmu. Ten by byl ovšem více paměťově náročný, a proto jsem jsem využil lepší a rychlejší implementaci pomocí zásobníku.

## 6.5 Hudební přehrávač

Mezi základní součásti multimediálních center neodmyslitelně patří také audio přehrávač, ať už v provedení jako přijímač internetového rádia, přehrávač MP3 nebo klasických CD. Přehrávání zvládne bezproblémů realizovat Java Media Framework rozšířená o FOBS, a to včetně internetových rádií. Rozhraní by mělo obsahovat správce kolekcí a seznamů skladeb. Aby nebylo přehrávání příliš fádní, bylo by vhodné ho doplnit buď vhodnou animací nebo analyzátozem. Tuto část jsem neměl již čas implementovat a zůstala pouze ve stavu návrhu.

---

<sup>2</sup>rekurzivní funkce je funkce, která volá sama sebe a obsahuje podmínku pro ukončení volání

# Kapitola 7

## Ovládání

Aplikace multimediálních center vyžadují možnost ovládání i pomocí jiných než standardních zařízení. Jako standardní zařízení se považuje klávesnice a myš, ovšem pro ovládání multimediálních center je vhodné využít například infračervené dálkové ovládání, ovládání po síti nebo například Bluetooth.

Pro zachytávání příkazů pomocí zpráv jsem definoval i zprávy pro ovládání a k nim třídy, které na základě uživatelských vstupů budou tyto zprávy předávat. Tím se docílila nezávislost modulů na zbytku systému, jednotlivé moduly reagují na signály od tříd snímajících vstupní zařízení. Zároveň je také zaručena univerzálnost a konfigurovatelnost. Třída pro klávesové ovládání má konfiguraci přímo implementovanou ve svém kódu. Výhodnější je však oddělení příkazů a k nim příslušejících kláves do konfiguračního souboru. Zde opět narážíme na špatný návrh rozhraní konfigurace.

### 7.1 Klávesnice

Ovládání myši běžným klikáním je již v systému implementováno využitím ovládacích prvků ze třídy `PictureButton`, která je potomkem třídy `Button`. Rozšířením nadtřídy `Controller` by bylo možné implementovat ovládání například pomocí gest myši.

Implementace ovládání pomocí klávesnice je poněkud obtížnější. Je třeba zajistit, aby byla `focusable` (označitelná) pouze jedna komponenta, která zachytává události klávesnice. Vyhodnocování je implementováno v modulu `SimpleKeyboardController`, který je implementací `KeyListener`. Moduly nadtřídy `Controller` přijímají zprávu `AddUnderControll`, která vyvolá metodu `addUnderControll` s parametrem zprávy obsahující komponentu, na které se má naslouchat například stiskům kláves.

Jakmile je zaregistrován `KeyListener`, začnou všechny stisky kláves vyvolávat metodu `keyPressed`. V této třídě se pak provádí identifikace klávesy pomocí metody `getKeyChar` nebo `getKeyCode` a následně se provede rozeslání zpráv pro všechny moduly nebo nadtřídy, které na toto tlačítko mají reagovat.



## Kapitola 8

# Testování

Cílem testování je objevit co možná nejvyšší počet softwarových chyb, nikoliv zajistit bezchybnost systému. Softwarovým testováním není možno zajistit naprostou bezchybnost systému, jednoduše z toho důvodu, že nelze reálně nasimulovat nekonečné množství vstupních hodnot a následně zkontrolovat nekonečné množství výstupních hodnot, otestovat všechny možné standardní a nestandardní situace a možnosti.

Metody softwarového testování dělíme obecně do dvou kategorií a to black box testování a white box testování. Tyto dvě kategorie zahrnují obecný inženýrský přístup k testování.

**Black box testování** (Testování metodou černé skřínky) je činnost, kdy testující nemá znalosti o tom, jak se testovaný systém chová uvnitř. Tester pouze předává vstupy, postupuje dle připravených scénářů a na konci porovná výsledek, zda se shoduje s očekáváním.

**White box testování** (Testování metodou bílé skřínky) nazýváme vždy, kdy tester musí nahlídnout dovnitř programu. Například při výběru vhodných vstupních dat.[\[15\]](#)

Při implementaci třídy `Module` bylo nutné zajistit její bezproblémovou funkcionalitu z hlediska přidávání a odebrání zpráv. Vytvořil jsem jednotkové testy pro testování s využitím testovacího nástroje TestNG [\[10\]](#), u něhož jsem využil možnost integrace ve vývojovém prostředí Eclipse. Testy jsou psány metodou Black box.

Testovány byly čtyři podstatné metody, `addActionList`, `removeFromActionList`, `isEnabledAction` a `getActionList`. První metoda složí k registrování zprávy, druhá k jejímu odebrání ze seznamu a třetí otestuje, je-li zpráva zaregistrována u daného modulu a dá se tedy zaslat. Protože je třída `Module` abstraktní, musel se vytvořit její potomek, který se následně testoval.

Pro testování metody `getActionList` byla vytvořena metoda `getActionListTest`, která kontroluje počet zaregistrovaných zpráv před testem. Další metoda pro testování `isEnabledActionTest` otestuje jak název napovídá metodu `isEnabledAction` a to tak, že otestuje tři správné varianty a jednu chybnou. Poslední test `removeFromActionListTest` testuje odebrání zpráv. Postupně se odebírají zprávy z původní trojice a testuje se jejich nepřítomnost i přítomnost mezi zbývajících zprávami.

# Kapitola 9

## Závěr

Při práci na této aplikaci jsem se více seznámil s programovacím jazykem Java a naučil se používat knihovnu pro multimédia JMF. Rozšířil jsem si také znalosti o grafickém rozhraní a jeho tvorbě pomocí knihovny Swing. Mezi neocenitelné zkušenosti patří praktické využití jednotkového testování. Testy mi pomohly odhalit několik chyb, kterých jsem se v průběhu vývoje dopustil. Také schopnost plánování je i u projektu takového, nepříliš značné, velikosti klíčová. Určit, do kdy musí být hotové jednotlivé části a následné dodržení je velmi náročné.

Projekt jsem vyvíjel na linuxové distribuci Arch Linux [4] vycházející z původní distribuce Slackware [5]. Pro vývoj v jazyce Java mám nainstalovány balíčky Java Developer Kit verze 6-2 a Java Runtime Environment verze 6u5-1. Pro řízení překladu jsem používal Apache Ant [8]. Jako vývojové prostředí mi sloužilo Eclipse SDK [12], s pluginy pro Subversion [11] a TestNG [10]. Subversion nebo-li SVN jsem využíval pro zálohování a také tvorbu verzí, ke kterým je možno se při chybách vracet.

### 9.1 Výsledky

Mým úkolem bylo navrhnout a vytvořit multimedialní aplikaci použitelnou pod systémem GNU/Linux. Systém je rozčleněn do modulů, které jsou na sobě nezávislé a umožňují snadné rozšíření a úpravy funkcí. Systém je intuitivně ovladatelný a má připraveno rozhraní pro využití více ovládacích zařízení, nejen pouze pomocí klávesnice a myši.

Z přehrávačů jsou zpracovány moduly pro prohlížení fotografií a přehrávání videa, který není zcela optimální. V knihovně JMF nebo jejím rozšíření FOBS se nachází zřejmě chyba nebo jsem je špatně použil. Video se přehrává v podstatně nižším rozlišení a také se mírně zasekává. Není-li ovšem umožněno přehrávat současně s videem zvuk, obsazením zvukového výstupu jiným programem, video se přehrává plynule.

### 9.2 Další vývoj

Aby byl systém kompletní podle specifikace uvedené v úvodu (1), bude nutné doplnit modul pro přehrávání hudebních médií včetně vhodného rozhraní. Dále je třeba vyřešit problém s přehráváním videa a rozšířit o možnost přehrávání DVD titulů.

Nyní je systém bez možnosti mít v konfiguraci uložené umístění CD/DVD mechaniky, a proto neumožňuje snadné přehrávání médií uložených na discích. Toto by bylo také velmi vhodné doplnit. Nutit uživatele, aby vždy mechaniku softwarově připojil, pak ji našel mezi adresáři, je silně neergonomické.

Další nedílnou součástí pro přehrávání filmů je zobrazování titulků. Tato funkce se dá snadno implementovat vytvořením nové nadtřídy, do které spadá také OSD<sup>1</sup>.

Důležitou součástí multimediálního centra je přehrávání DVD a práce s televizní kartou. Přehrávání filmů je již implementováno, avšak potřebuje rozšířit o práci s DVD. U televizní karty však nastávají komplikace s jejím ovládním. Pro jazyk C existuje několik variant pro změnu kanálů a celkovou práci s kartou, pro Javu zatím nic takového neexistuje a zřejmě bude nutné využít právě tyto ovládací knihovny.

Rád bych tento program dále vyvíjel a využíval ho v naší domácnosti. Pro tento účel mám již připojenou televizi a reproduktorovou soustavu ke stolnímu PC. Uvažuji nad koupí dálkového ovládní a televizní karty.

---

<sup>1</sup>On-Screen Display představuje text nebo grafiku překrývající z části obraz. Nejznámější OSD je například číslo zobrazované na televizní obrazovce při přepínání kanálů.

# Literatura

- [1] Domovská stránka Elisa.  
URL [⟨http://elisa.fluendo.com⟩](http://elisa.fluendo.com)
- [2] Domovská stránka GeeXboX.  
URL [⟨http://www.geebox.org/en/index.html⟩](http://www.geebox.org/en/index.html)
- [3] Domovská stránka LinuxMCE.  
URL [⟨http://www.linuxmce.org⟩](http://www.linuxmce.org)
- [4] Domovská stránka linuxové distribuce Arch Linux.  
URL [⟨http://www.archlinux.org/⟩](http://www.archlinux.org/)
- [5] Domovská stránka linuxové distribuce Slackware.  
URL [⟨http://www.slackware.com/⟩](http://www.slackware.com/)
- [6] Domovská stránka Media Center.  
URL [⟨http://www.microsoft.com/cze/windows/xp/mediacenter/default.aspx⟩](http://www.microsoft.com/cze/windows/xp/mediacenter/default.aspx)
- [7] Domovská stránka MythTV.  
URL [⟨http://www.mythtv.org⟩](http://www.mythtv.org)
- [8] Domovská stránka projektu Apache Ant.  
URL [⟨http://ant.apache.org/⟩](http://ant.apache.org/)
- [9] Domovská stránka projektu dom4j.  
URL [⟨http://dom4j.org/⟩](http://dom4j.org/)
- [10] Domovská stránka projektu TestNG.  
URL [⟨http://testng.org/doc/⟩](http://testng.org/doc/)
- [11] Domovská stránka verzovacího nástroje Subversion.  
URL [⟨http://subversion.tigris.org/⟩](http://subversion.tigris.org/)
- [12] Domovská stránka vývojového prostředí Eclipse.  
URL [⟨http://www.eclipse.org/⟩](http://www.eclipse.org/)
- [13] Encyklopedie WikipediA - Extensible Markup Language.  
URL [⟨http://cs.wikipedia.org/wiki/XML⟩](http://cs.wikipedia.org/wiki/XML)
- [14] Encyklopedie WikipediA - Java.  
URL [⟨http://cs.wikipedia.org/wiki/Java⟩](http://cs.wikipedia.org/wiki/Java)
- [15] Encyklopedie WikipediA - Testování software.  
URL [⟨http://cs.wikipedia.org/wiki/Testovani\\_software⟩](http://cs.wikipedia.org/wiki/Testovani_software)

- [16] Encyklopedie WikipediA - Windows Media Center.  
URL [⟨http://en.wikipedia.org/wiki/Windows\\_Media\\_Center⟩](http://en.wikipedia.org/wiki/Windows_Media_Center)
- [17] Java<sup>TM</sup> 2 Platform, Standard Edition, v 1.4.2, API Specification.  
URL [⟨http://java.sun.com/j2se/1.4.2/docs/api/index.html⟩](http://java.sun.com/j2se/1.4.2/docs/api/index.html)
- [18] Objektová analýza, návrh a programování.  
URL [⟨http://objekty.vse.cz/Objekty/Vzory-Singleton⟩](http://objekty.vse.cz/Objekty/Vzory-Singleton)
- [19] Přehrávání a střih videozáznamů v jazyce JAVA.  
URL [⟨http://www.kiv.zcu.cz/~lobaz/mhs/semestralky/2006/video\\_Java/⟩](http://www.kiv.zcu.cz/~lobaz/mhs/semestralky/2006/video_Java/)

# Dodatek A

## Přílohy

**Příloha 1:** Datový nosič CD se zdrojovými kódy, programovou dokumentací, konfiguračním souborem a elektronickou kopií této technické zprávy.