

UNIVERZITA PARDUBICE
DOPRAVNÍ FAKULTA JANA PERNERA

BAKALÁŘSKÁ PRÁCE

2009

Václav Lenoč

UNIVERZITA PARDUBICE
DOPRAVNÍ FAKULTA JANA PERNERA

KOMPRESSE ZVUKOVÝCH SIGNÁLŮ

Václav Lenoč

Bakalářská práce

2009

Univerzita Pardubice
Dopravní fakulta Jana Pernera
Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě
Akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Václav LENOCH
Studijní program: B3709 Dopravní technologie a spoje
Studijní obor: Dopravní infrastruktura-Elektrotechnická zařízení
v dopravě
Název tématu: Komprese zvukových signálů

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

Seznamte se s principy komprese zvukových signálů.

Navrhněte a realizujte jednoduchý audio kodér a jemu odpovídající dekodér.

1. Seznamte se s principy bezztrátové komprese zvukových signálů (obecný princip)
2. Porovnejte vybrané bezztrátové komprese zvukových formátů z hlediska účinnosti a rychlosti komprese i vzhledem k běžným kompresním programům (RAR, ZIP)
3. Naprogramujte v libovolném prostředí (C, Matlab) jednoduchý kodér (případně i dekodér) bezztrátového kodeku

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

<http://www.zive.cz/default.aspx?article=131549>

<http://members.home.nl/w.speek/comparison.htm>

<http://web.inter.nl.net/users/hvdh/lossless/lossless.htm>

<http://cs.wikipedia.org/wiki/Kodek>

<http://www.root.cz/clanky/pcx-prakticky-implementace-komprimace-rle/>

Vedoucí bakalářské práce:

Ing. Jan Pidanič

Katedra elektrotechniky, elektroniky a zabezpečovací
techniky v dopravě

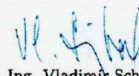
Datum zadání bakalářské práce: **23. února 2009**

Termín odevzdání bakalářské práce: **1. června 2009**



prof. Ing. Bohumil Culek, CSc.
děkan

L.S.



prof. Ing. Vladimír Schejbal, CSc.
vedoucí katedry

V Pardubicích dne 23. února 2009

Prohlašuji:

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 25. března 2009

Václav Lench

Poděkování

Děkuji vedoucímu bakalářské práce panu Ing. Janu Pidaničovi za jeho odborné vedení a ochotu poskytovat konzultace.

Dále děkuji všem, kteří mi při vytváření bakalářské práce vyšli vstříc nebo mi pomohli svými cennými radami. V neposlední řadě patří poděkování rodině a spolužákům, kteří mě v průběhu celého studia na vysoké škole podporovali.

Souhrn

Tato bakalářská práce se zaměřuje na bezztrátovou kompresi zvukových signálů, metody komprese, popis a srovnání vybraných programů. Práce se zabývá na měření rychlosti komprese a dekomprese a velikost jejich komprese. Dále v kapitole 5. je popsáno vytvoření kodéru a dekodéru v jazyce C#.

Klíčová slova

bezeztrátová komprese zvuku, dekomprese, kódování

Title

Compression sound signals

Abstract

This bachelor work is focusing to the lossless compression of audio signals, compression methods, description and comparison of the selected programmes. This paper deal with compression and decompression speed measurement and the compression measurement. Also in the Chapter 5. there is a description of code of the codec & encoder in the C# language.

Keywords

compression sound signals, decompression, encoding

Obsah:

| | |
|--|-----------|
| Úvod | 9 |
| 1. Seznamení s principy bezeztrátové komprese zvukových signálů | 10 |
| 1.1. Porovnání komprese | 10 |
| 1.1.1. Poměr komprimace..... | 10 |
| 1.1.2. Faktor komprimace | 10 |
| 1.1.3. Čas potřebný pro komprimaci | 11 |
| 1.2. Postupy komprimace..... | 11 |
| 1.3. Rozdělení bezeztrátových algoritmů | 11 |
| 1.3.1. Transformace..... | 11 |
| 1.3.2. Statistické algoritmy | 12 |
| 1.3.3. Slovníkové algoritmy..... | 12 |
| 1.3.4. Ostatní algoritmy..... | 13 |
| 2. Popis vybraných bezeztrátových formátů | 14 |
| 2.1. Free Lossless Audio Codec | 14 |
| 2.1.1. Způsob komprese | 14 |
| 2.1.2. Parametry | 15 |
| 2.1.3. Srovnání | 15 |
| 2.2. Apple Lossless | 15 |
| 2.2.1. Způsob komprese | 15 |
| 2.2.2. Srovnání | 16 |
| 2.3. Monkey's Audio | 16 |
| 2.3.1. Způsob komprese | 16 |
| 2.3.2. Srovnání | 17 |
| 2.4. WMA Lossless | 17 |
| 2.4.1. Způsob komprese | 17 |
| 2.5. ZIP..... | 17 |
| 2.5.1. Způsob komprese a možnosti..... | 18 |
| 2.6. RAR..... | 19 |
| 2.6.1. Způsob komprese a možnosti..... | 19 |

| | |
|--|-----------|
| 3. Porovnání vybraných bezztrátových kompresních zvukových a běžných kompresních programů | 21 |
| 3.1. Vybrané bezztrátové komprimační programy pro porovnání..... | 21 |
| 3.2. Konfigurace počítače..... | 21 |
| 3.3. Postup měření | 21 |
| 3.3.1. Měření komprese..... | 22 |
| 3.3.2. Měření rychlosti | 22 |
| 3.4. Výběr zvukového záznamu | 23 |
| 3.5. Výsledky měření..... | 24 |
| 3.6. Zhodnocení výsledků | 32 |
| 4. Ověřování bezztrátové komprese | 34 |
| 4.1. Ruční porovnání | 34 |
| 4.2. Automatické ověřování..... | 34 |
| 5. Vytvoření jednoduchého bezztrátového kodéru a dekodéru | 35 |
| 5.1. Run-length encoding | 35 |
| 5.2. Řešení algoritmu | 35 |
| 5.3. Provedení..... | 36 |
| 5.4. Ověření funkce | 37 |
| Závěr..... | 39 |
| Použitá literatura | 40 |
| Seznam obrázků | 44 |
| Seznam tabulek | 45 |

ÚVOD

Kompresi zvukových signálů se používá pro ukládání a přenos zvukových záznamů. Při přenosech v reálném čase je ji možno užít ke zmenšení datového toku. Výsledkem je zkrácení doby potřebné pro přenos, čehož lze využít pro přenos dat v telekomunikačních či datových sítích. Další významné použití je při archivaci a záloze dat.

Ke kompresi se využívají matematické algoritmy vhodné pro zmenšení objemu dat. Novým uspořádáním nebo odstraněním nadbytečných informací se zvýší entropie (průměrná informace na jeden znak). Z hlediska principu dělíme komprese na dva typy:

Bezeztrátová komprese – její hlavní výhodou je, že po dekomprimaci získáme stejná data jako před kompresí (nedochází ke ztrátě). Lze ji obráceným algoritmem vrátit do původní podoby, originálu, který se musí shodovat. Účinnost není taková jako u ztrátové komprese.

Ztrátová komprese – při kompresi zde dochází ke ztrátě určitého množství dat, které jsou pro další využití irelevantní. Při ztrátové kompresi se využívá nedokonalosti lidských smyslů, např. sluchu: lidské ucho neslyší celé spektrum zvukového signálu stejně. Proto dochází k oříznutí částí frekvenční charakteristiky a ke ztrátě postraních částí spektra záznamu. Stupeň komprese je nepřímo úměrný kvalitě záznamu. Nejčastěji se používá pro zvukové, obrazové a video záznamy.

Stěžejní priorita bude směřována na bezeztrátovou kompresi zvuku vybraných kompresních programů. V druhé řadě budou popsány typy formátů využívající různé matematické algoritmy. Bude provedeno měření rychlosti komprese, dekomprese a stupeň komprimace. Bezeztrátové kodeky budou porovnány s obecnými archivačními programy RAR a ZIP. V kapitole 6 bude vytvořen komprimační program pomocí programovacího jazyka C#.

1. SEZNAMENÍ S PRINCIPY BEZEZTRÁTOVÉ KOMPRESY ZVUKOVÝCH SIGNÁLŮ

Pro bezeztrátovou kompresi využíváme matematické algoritmy, které umožňují zmenšení velikosti dat a zpětnou rekonstrukci zkomprimovaných dat do originální podoby, což není možné u ztrátové komprese dat. Bezeztrátová komprese umožňuje zachování dat v originální kvalitě, v níž byly vytvořeny.

1.1. Porovnání komprese

Pro kompresi se používají různé matematické algoritmy, z kterých vzniknou jednotlivé kompresní metody. Každá z těchto metod je výhodná pro určitý formát souboru. Je nutné zvolit správný formát, aby byla komprimace dat co nejefektivnější. Každá komprimační metoda je jinak časově náročná a efektivní. Jednotlivé algoritmy pro kompresi se srovnávají podle několika hledisek uvedených níže.

1.1.1. Poměr komprimace

Vzorec 1.1 nám ukazuje poměr, v jakém byla provedena komprese dat. Například při komprimaci 30 MB souboru vznikne 15 MB soubor. Znamená to, že poměr je $30/15 = 2$ a komprimační poměr je $2 : 1$. Pokud chceme tento poměr přepočítat vůči původní velikosti na jakou úroveň se zmenší $1 - 15/30 = 0,5$. Po vynásobení 100 dostaneme hodnotu 50% původní velikosti. Může se stát, že při nevhodném algoritmu dojde ke zvětšení komprimovaného souboru.

$$Pomer\ komprimace = \frac{vstupní\ velikost}{výstupní\ velikost} \quad (1.1)$$

1.1.2. Faktor komprimace

Jedná se o převrácenou hodnotu komprese. Například: vyjde-li číslo 0,5, znamená to, že výstupní soubor po komprimaci bude mít velikost 50% originálního souboru.

$$Faktor\ komprimace = \frac{výstupní\ velikost}{vstupní\ velikost} \quad (1.2)$$

1.1.3. Čas potřebný pro komprimaci

Jedním z rozhodujících faktorů při výběru vhodné komprese je čas potřebný pro kompresi či dekompresi. Zpravidla platí, že čím je účinnější komprimace, tím je časově náročnější. Zároveň záleží na zvolených typech kompresních algoritmů a komprimovaném typu souborů (audio, video, text).

1.2. Postupy komprimace

Ke způsobu komprimace se přistupuje dvěma různými metodami. První metoda postupuje při kódování informace po celém bloku sousedních bajtů. Uvádí se jako blokové metody (block mode). Načítá vstupní informační blok po bloku a každý blok se zakóduje samostatně.

Druhý možný přístup ke komprimaci je tzv. streaming mode. V tomto pojetí je programem načten jeden nebo více bajtů informace, které jsou dále zpracovány (zakódovány) v reálném čase jako souvislý tok dat a tento postup se uplatňuje až do konce komprimovaného souboru. [1]

1.3. Rozdělení bezeztrátových algoritmů

U většiny komprimačních programů se používá více než jeden algoritmus. Toho využívají některé programy pro dosažení vyššího stupně komprese. Vzhledem k rozdílnostem jednotlivých typů souborů se liší svoji strukturou a složitostí. Proto algoritmy rozdělujeme do několika skupin, které se od sebe liší metodou, se kterou zpracovávají dané data.

- Transformace
- Statistické algoritmy
- Slovníkové algoritmy
- Další

1.3.1. Transformace

Tyto algoritmy ve skutečnosti nic nekomprimují, pouze modifikují data tak, aby se dala lépe zkomprimovat. [9] Mohou přitom změnit hodnoty prvků a mohou obrátit pořadí prvků. Tento typ algoritmu je založen na lineární transformaci, která převede data z časové oblasti vzorků do frekvenční oblasti. Také je podmínkou, že ke každé

transformaci musí existovat transformace inverzní, která bude schopna obnovit původní data.

- Burrows-Wheeler transformation (BWT) [12]
- Weighted frequency count (WFC) [33]
- Distance cosiny (DC) [34]
- Move to front I (MTF I) – česky „přesuň na začátek“ [13]
- Move to front II (MTF II)
- Inverse frequency coding (IF) [35]

1.3.2. Statistické algoritmy

Snaží se určitým způsobem předvídat, jaké znaky budou v souboru dat následovat. Pro znaky s vyšší pravděpodobností výskytu vyhradí algoritmus kratší informaci pro jejich zapsání, pro znaky s nižší pravděpodobností výskytu vyhradí naopak delší informaci pro jejich zapsání. [9]

Statistické metody [14] dále dělíme na metody se statickým modelem (model slouží pro vypočítávání pravděpodobnosti výskytu znaků) a metody s adaptivním modelem.

Metody se statickým modelem vytvoří před komprimací dat určitý model podle něhož zkomprimují celý soubor dat, zatímco metody s adaptivním modelem průběžně model aktualizují. Obecně se dá říci, že metody se statickým modelem bývají dvouprůchodové a metody s adaptivním modelem jednopřechodové. [9]

- Shannon-Fanovo kódování [14]
- Huffmanovo kódování [15]
- Aritmetické kódování [14]
- Range coding (RC) [16]
- ACB [17]
- Prediction by partial match (PPM) [18]

1.3.3. Slovníkové algoritmy

Algoritmy v této skupině vytvářejí v průběhu komprimace slovník na základě dat již zkomprimovaných, v němž se pak snaží najít data, která se teprve mají komprimovat. Pokud jsou data nalezena ve slovníku, algoritmus zapíše pozici dat ve slovníku místo samotných dat. [9] Při komprimaci textu se vybírají celé řetězce

znaků. Tyto algoritmy se dále rozdělují na statické kódové slovníky a dynamické kódové slovníky.

Příklad algoritmu LZ77: řetězec „leze po železe“ zkomprimuje jako: „leze po že [10,4]“. Z čísel se při dekompresi vyčte, že se využijí 4 znaky z přecházejících deseti.

- Lempel-Ziv 77 (LZ77) - PKZip, WinZip, LHArc, PNG, gzip a ARJ [19]
- Lempel-Ziv 78 (LZ78) [20]
- Lempel-Ziv-Welch 84 (LZW84) - podobný LZ78, byl velmi populární, ale také patentovaný, GIF, TIFF [14]
- LZMA - zip, pdf [21]

1.3.4. Ostatní algoritmy

Algoritmy, které nelze zařadit do výše uvedených skupin.

- Run length encoding (RLE) [22]
- Golomb coding [25]
- Potlačení nul [23]
- Bitové mapy [23]
- Půlbajtové kódování [23]
-

RLE komprimuje opakující se znaky jdoucí po sobě tak, že zapíše znak a za něj počet opakování. Předchůdcem této metody je potlačení nul, který nekomprimoval všechna data pouze mezery a nuly.

Bitové mapy jsou kombinací mezi RLE a potlačení nul. Jejich využití najde uplatnění u dat, kde se ve velkém počtu vyskytuje jeden stejný znak, který se komprimuje.

Půlbajtové kódování ukládá jeden bajt do půlbajtu, druhou polovinu doplní z následujícího bajtu. Efektivní je pouze u malého počtu znaků, například v bankovníctví, kde se nachází jen čísla a druh měny.

2. POPIS VYBRANÝCH BEZEZTRÁTOVÝCH FORMÁTŮ

Pro popis bylo vybráno šest nejznámějších bezetrátových kompresních formátů, z toho čtyři zvukové a dva univerzální komprimační programy.

- Free Lossless Audio Codec
- Apple Lossless
- Monkey's Audio
- WMA Lossless
- ZIP
- RAR

2.1. Free Lossless Audio Codec

Známý pod zkratkou FLAC, což v překladu znamená bezetrátový zvukový kodek. Jedná se o otevřený software, který není chráněn žádným patentem. To umožňuje jednodušší šíření a implementaci tohoto kodeku do přehrávačů a přenosných zařízení. Od 29. ledna 2003 výrobce Xiph.Org Foundation začal FLAC přidávat k formátům Vorbis, Ogg, Speex. FLAC je podporován na platformách Unix, Linux, * BSD, Solaris, Max OS X, BeOS, Windows a OS/2.

2.1.1. Způsob komprese

Tento kodek využívá lineární predikce (Lineární prediktivní kódování LPC) [26], která zajišťuje konverzi zvukových vzorků do série malých čísel (známé jako reziduály). Ty jsou dále efektivně uloženy pomocí Golom-Ricova kódování [25]. Pro zajištění bitové přesnosti nepoužívá vzorky s plovoucí desetinnou čárkou. [2] Dále je využíváno RLE [22], které je ve skutečnosti efektivní pouze do 8 bitového rozlišení, neboť u něj dochází při zápisu k tomu, že následuje po sobě několik shodných hodnot vhodných pro kompresi. Ty se mohou vyskytovat i u 16 či 32 bitového, ale pravděpodobnost výskytu je daleko menší. V tomto případě by komprese nebyla dosažena. Z tohoto důvodu je RLE pro 16 a více bitů nevyužitelný. Nepřehlédnutelným faktorem možnosti komprese je uspořádání bajtů v záznamu, které se liší podle počtů zvukových kanálů. Při výskytu více než jednom kanálu se shodnost dvou jdoucích znaků po sobě nevyskytuje.

2.1.2. Parametry

Vzorkovací frekvence se pohybuje v mezích 1 Hz a 1 048 570 Hz. Při podpoře 1 až 8 zvukových kanálů a to při 4 až 32 bitech. Mezi hlavní výhody tohoto formátu patří malá hardwarová náročnost a podpora streamování (datový tok přijímaný ze sítě se neukládá na disk, ale je přímo přehráván). Formát je podporován celou řadou přenosných přehrávačů. Na českém trhu nalezneme například Cowon audio. U přehrávačů značky iPod a iRiver jen za podpory neoficiálního firmwaru Rockbox.

2.1.3. Srovnání

FLAC dosahuje vysoké stupně komprese zvuku na rozdíl od obecných programů jako jsou WinZip nebo WinRAR, které slouží ke komprimaci libovolných dat. Zip dosahuje velikosti komprimace zpravidla 94 – 90 % u zvukových záznamů, oproti kodeku FLAC, který má míry komprimace 45 – 70 %. FLAC má oproti nim vyšší rychlost a je více rozšířen. Stupeň dosažené komprese zvukového souboru se u jednotlivých verzí liší. Při srovnání novější verze 1.1.4 dosahuje výrazně lepších výsledků komprese, než verze 1.1.2 nebo starší.

2.2. Apple Lossless

Také známý jako Apple Lossless Encoder (ALE) nebo jako Apple Lossless Audio Coder (ALAC). Tento audio kodek vyvinutý společností Apple Computer byl představen 28. dubna 2004 jako součást QuickTime verze 6.5.1. Hudba kódovaná pomocí Apple Lossless je uložena pomocí kontejneru MP4. Jedná se o multimedialní kontejner definovaný standardem ISO/IEC 14496-14:2003. Častěji je znám pod názvem MPEG-4. Užívá příponu souborů .m4a. Kodek nepoužívá metodu DMR (Digital Rights Management - Správa digitálních práv), ale vzhledem k typu používaného kontejneru je možné DRM aplikovat na ALAC podobně jako u ostatních souborů využívajících kontejnery QuickTime.

2.2.1. Způsob komprese

Užívá lineární predikci [26] podobně jako ostatní bezztrátové audio kodeky (např. kodek FLAC nebo Shorten). Využívá adaptabilních metod a Golomb kódování. Dekodér analyzovali David Hammerton a Cody Brocius bez jakékoli dokumentace 5. března 2005. Dekodér byl vytvořen v programovacím jazyku C. [27]

2.2.2. Srovnání

Od ostatních formátů se Apple Lossless liší nízkými nároky na dekompresi a proto je užíván u zařízení s omezeným zdrojem energie jako například iPod.

2.3. Monkey's Audio

Jedná se o bezztrátový kodek, který využívá pro ukládání koncovkou APE. Má rychlé kódování, ale náročnější dekódování. Při nastavení na nejmenší kompresi zvládne Monkey's přehrát přenosné zařízení s použitím firmware Rockbox. Při přehrávání většího stupně komprese by bylo zapotřebí výkonnějšího procesoru v podporovaných přenosných zařízeních. Podporuje tagy typu ID3 nebo vlastní formát APE. Podporuje pouze 2 kanály, což je méně než FLAC a ALAC. Účinnost komprese je mezi bezztrátovými kodeky největší, průměrně kolem 50 %, což je však na úkor rychlosti. Platí, že čím větší je nastavená komprese, tím je doba potřebná k této kompresi delší.

Monkey's audio využívá detekci chyb a připojuje redundantní CRC. Slouží to k tomu, aby byla zajištěna bezchybná dekomprese dat. Zdrojový kód je dostupný pro další úpravy.

Hlavní podpora je OS Windows, Linux podporuje pouze okrajově. SuperMMX uvolnil neoficiální podporu začátkem roku 2005. Rovněž zahrnuje plugin umožňující přehrávat v programech XMMS a Beep Media Player pro OS Unix a Linux. Zahrnuje podporu pro Mac OS X stejně jako Linux na PowerPC a SPARC architektury [29].

2.3.1. Způsob komprese

Tento kodek využívá optimálního ukládání kanálů, které je založeno na průměrování kanálů a následném výpočtu rozdílů kanálů. Dále následuje předpovídání průběhu dat pomocí využití lineární algebry. Vypočte se předpokládaná hodnota, která se porovnává s původním zvukovým záznamem. Poté se vyhodnotí, zdali byla předpověď užitečná nebo ne. Dle toho se předpověď přizpůsobuje nahoru nebo dolů. Následuje výpočet rozdílu ze skutečného záznamu a předpovědi. Snahou je ukládat co nejmenší hodnotu, která se zakóduje pomocí Riceova kódu [25]. [28]

2.3.2. Srovnání

Komprimace Monkey's audio je přibližně o 2 % lepší než FLAC a ALAC. Ve srovnání rychlosti komprese je přibližně o 55 % pomalejší než FLAC, rychlost je závislá na nastavení komprese. Podporuje pouze 2 zvukové kanály.

2.4. WMA Lossless

Je zkratka Windows Media Audio 9. Jedná se o zvukový bezztrátový kodek uvedený na trh firmou Microsoft na začátku roku 2003. Jde o nejlepší ze všech kodeků Microsoftu. Pro ukládání používá stejnou koncovku jako ztrátový .WMA. Podporuje až 6 nezávislých kanálů, zaznamenaných až do 24 bit / 94 kHz.

Technická podpora OS Windows, Windows mobil, Xbox 360, NWZ-A a NWZ-S, Toshiba Gigabeat S a V modely.

2.4.1. Způsob komprese

Kodek nabízí kontrolu dynamického rozsahu, používající maximální a průměrné zvukové amplitudy, které byly vypočítané během kódovacího procesu. Tato funkce se jmenuje Quiet Mode (ztišený režim) a je podporován ve Windows multimediálním přehrávači verze 9 a vyšších. Uživatelé mohou slyšet plný dynamický rozsah zvuku. Střední rozsah zvukové amplitudy sahá až 12 dB nad průměr, malý rozsah se pohybuje do 6 dB nad průměrem [30]. WMA Lossless využívá variable bitrate (VBR).

2.5. ZIP

Jedná se o známý komprimační program, který slouží k bezztrátové kompresi dat (archivaci). Byl vytvořen Philem Katzem pro PKZIP v polovině osmdesátých let. Název ZIP jako význam rychlosti, na rozdíl od pomalejšího ARC. Dnes s ním pracuje celá řada dalších programů, jako je RAR, WinRAR, WinZIP, BOMArchiveHelper, StuffIt, KGB Archiver, PicoZip, Info-ZIP, IZARC, 7-Zip, ALZip, TUGZip, PeaZIP, ZipGenius a ACE. Většina programů dnes dosahuje lepší komprese a lepší funkce, než ZIP nenabízí. Je možné se setkat s jinou koncovkou než ZIP. Příkladem mohou být dokumenty OpenOffice.cog, Office Open XML, Java JAR soubory, Id Software (.pk3/ .pk4) a Winamp skin, Mozilla balíčky .xpi. Ze světa

mobilních telefonů Nokia a Sony Ericsson používají zip pro kompresi témat (.nth a .thm).

2.5.1. Způsob komprese a možnosti

ZIP je poměrně jednoduchý archivní formát, který komprimuje každý soubor zvlášť. To umožňuje použití různých algoritmů pro kompresi, na jednotlivé soubory. V případě, že se jedná o velké množství malých souborů, nedosáhne takové úrovně komprese, jako kdyby byly komprimovány pouze jednou metodou od začátku až do konce (GZIP). Pro komprimaci se užívá Deflerův algoritmus, který se skládá ze slovníkového algoritmů LZ77 [19] a LZ78 [20]. Ty zajišťují zkrácení opakujících se částí tak, že „nové“ části jsou jednou zapsány a na ostatní místa se zapíše pouze jejich přiřazení. Po provedení první části se aplikuje statický algoritmus (Huffmanovo kódování) [15]. Konvertuje znaky vstupních souborů různých případně stejných délek do bitových řetězců. Znakům, které se vyskytují v řetězcích s největší četností, je přiřazen nejkratší bitový řetězec.

Originální ZIP má omezení velikosti (dekomprimovaná velikost souboru, celkové velikosti archivu a komprimovaná velikost souboru) 4,2 GB. Pro verzi 4.5 uvedla PKWARE ZIP64, který posouvá tyto hranice.

Od září 2007 ZIP obsahuje podporu k tomu, aby ukládal jméno souboru UTF-8 a kompatibilitu Unicode.

Verze komprimačního programu a jejich vylepšení:

- 2.0: Soubory mohou být komprimované metodou DEFLATE.
- 4.5: Přidáno 64-bit ZIP formát.
- 5.0: DES, 3DES, RC2, RC4 podpora pro šifrování
- 5.2: RC2-64 podpora pro Encryption.
- 6.1: Přidán certifikát uložení.
- 6.2.0: Přidáno centrální směrodatné šifrování.
- 6.3.0: Přidáno Unicode (UTF 8) ukládání jména souboru. Rozšířený seznam podporovaného hašé, komprese, šifrovacích algoritmů.
- 6.3.1: Opravený standardní hašé vhodného pro
– SHA- 256/384/512.
- 6.3.2: Přidaná kompresní metoda 97 (WavPack).

2.6. RAR

Další ze známých archivačních programů je RAR. Byl vyvinutý ruským programátorem Jevgenijem Rošalem, proto pojmenování **R**oshal **A**Rchive. Je možno se setkat také s koncovkou rev. Nejčastější program pro manipulaci s tímto typem souborů je WinRAR, ten zvládá jak komprimaci, tak dekomprimaci. Později byly uvolněny algoritmy pro dekompresi, které umožňuje tyto archivy dekomprimovat i v jiných programech.

2.6.1. Způsob komprese a možnosti

RAR dosahuje velmi dobrých výsledků stupně komprese a také poměru rychlosti k účinnosti komprese. Je podporován řadou operačních systémů: Windows, Linux, Unix, Macintosh, IBM OS/2, x64 Linux, Solaris, Irix 6.5, MorphOS a Pocket PC.

- Umí navíc vytvářet formát ZIP a číst archivy dalších konkurenčních programů: 7-Zip, ISO, ARJ, ACE, TAR, CAB, LZH, GZIP, JAR, BZ2 a UUE.
- Vícesvazkové archivy, ty mají pak koncovku .rar, .r00, .r01.
- Archiv lze rozdělit na díly, výhodné při archivaci na menší záznamové zařízení než je komprimovaný soubor.
- Data lze vybavit kontrolními CRC tak, aby po případném poškození bylo možné chybné místo dopočítat beze ztráty informace.
- Archiv lze doplnit o údaje, kdo jej vytvořil (obdoba digitálního podpisu).
- Možnost automatického generování jména podle data.
- Vytvoření samorozbalujících archivů.
- Využití více jádrových počítačů.

Vývojáři dělí RAR na několik vývojových částí:

- RAR (originál)
- RAR2
- RAR3 (aktuální) - WinRAR 2.9 a WinRAR 3.00 několik změn:
 - Možnost zašifrování pomocí AES se 128-bit, zamezení neoprávněné manipulace.
 - Šifrování obou souborových dat a záhlaví souboru.

- Vylepšený komprimační algoritmus použitím 4 MB velikostí slovníku, Dmitry Shkarin's PPMII algoritmus pro textová data.
- Podpora velkých archivních souborů větších než 4 GB a Unicode název.
- S výjimkou starší verze ZIPu archivační programy komprimují soubor jako celek.
- RAR je také implementován do obrázkového formátu .jpeg, kde zajišťuje kompresi.

3. POROVNÁNÍ VYBRANÝCH BEZEZTRÁTOVÝCH KOMPRESNÍCH ZVUKOVÝCH A BĚŽNÝCH KOMPRESNÍCH PROGRAMŮ

Vzhledem k tomu, že různé bezeztrátové kompresní programy užívají rozdílné postupy, liší se tím rychlost a stupeň komprese. Zároveň by bylo možno sledovat i další parametry, jako je náročnost na výkon hardwaru a počet zvukových kanálů, ovšem tato hlediska jsou pro tuto práci marginální.

Rychlost komprimačního programu je závislá na nastavení stupně komprese. Čas je rozdílný pro kompresi a dekompresi, proto je nezbytné porovnání v obou případech. Komprimace se ve většině případů provede pouze jednou na rozdíl od dekomprese, která se může vyskytovat častěji (např. přehrávání hudebních souborů v mp3 přehrávačích).

3.1. Vybrané bezeztrátové komprimační programy pro porovnání

Pro porovnání byly vybrány již dříve popisované komprimační programy. Zde jsou uvedeny včetně přípon:

- FLAC – 1.7.1 (.flac)
- APPLE LOSSLESS – XRECODE 2.39 (.m4a)
- MONKEY'S - 4.01 (.ape)
- WMA losless – XRECODE 2.39 (.wma)
- RAR – WinRAR 3.80 (.rar)
- ZIP - WinZIP 12.0 (.zip)

3.2. Konfigurace počítače

Pro měření bude použit počítač ASUS A7Tseries, AMD Turion(™) 64 X2 Mobilní Technologie TL-52 1,61 GHz duol-core, 803 MHz, 1 GB DDR2 642 MHz RAM, HDD FUJITSU MHV2120BH PL (SATA150, NCQ, 8MB cash) čtení 30,44 MB/S, Windows XP SP2.

3.3. Postup měření

Ovlivňujícími parametry měření jsou velikost komprese, rychlosti komprimace a dekomprimace. Měření se provede s několika rozdílnými typy audio záznamu,

příčemž získané hodnoty budou graficky znázorněny. V grafu bude zobrazena závislost komprese na rychlosti.

3.3.1. Měření komprese

Vzhledem k tomu, že se jedná o bezztrátovou kompresi, bude vstupní soubor v nekomprimovaném WAV formátu. Ten bude ověřen, zda nebyl vytvořen ze ztrátové komprese, z důvodu zajištění kvalitního porovnání. Při použití již ztrátově komprimovaného zvukového záznamu by byla komprese rozdílná. Poté se provede komprimace do daného formátu při specifickém nastavení komprese, při které se bude měřit čas komprimace. Možnosti nastavení komprese jsou uvedeny v tabulce 1.

| program | verze | možnosti nastavení komprese |
|----------------|--------------|---|
| FLAC | 1.7.1 Etree | 0, 1, 2, 3, 4, <u>5</u> , 6, 7, 8. |
| Apple Lossless | Xrecode 2.39 | |
| Monkey's Audio | 4.01 | fast, normal, high, extra high, insane. |
| WMA Lossless | Xrecode 2.39 | |
| WinZip | 12.0 | SuperFast, Enhanced Deflate, bzip2, LZMA, PPMd, best, Zip 2 |
| WinRAR | 3.80 | Nejrychlejší, Rychlá, Normální, Dobrá, Nejlepší |

Tab. 1 - Možnosti nastavení komprese

Pro výpočet komprese potřebujeme velikost již zkomprimovaného a originálního souboru.

$$komprese = \frac{\text{komprimovaná velikost}}{\text{velikost wavu}} \cdot 100 \quad [\%] \quad (3.1)$$

Následuje obrácený proces dekomprese, který převede soubor do původního formátu wav. U tohoto procesu měříme čas dekomprese. Díky opětovnému převedení souboru na formát WAV můžeme porovnat velikost i obsah konečného a počátečního souboru wav. Tyto hodnoty se musí naprosto shodovat. Z toho vyplývá, že během komprimace nedošlo ke změně a výsledkem je bezztrátová komprimace.

3.3.2. Měření rychlosti

V případě komprimace i dekomprimace se jedná o ryze matematický proces. Rychlost komprimace a dekomprimace je ovlivněna typem použitého algoritmu a rychlostí procesoru počítače. Výsledná rychlost je ovlivněna frekvencí sběrnice a rychlostí pevného disku pro čtení i zapisování.

Aby se dosáhlo lepší možnosti porovnání, je nutné čas přepočítat na *skutečný čas*. Je vyjádřen poměrem času délky skladby a času procesu.

$$\text{skutečný čas} = \frac{\text{délka skladby}}{\text{čas procesu}} \quad (3.2)$$

3.4. Výběr zvukového záznamu

Je známo, že ztrátové kompresní metody dosahují velké komprese tím, že utlumí (oříznou) vysoké frekvence. Například při pohledu na frekvenční spektrum formátu MP3 je vidět, že vysoké frekvence od 15 kHz jsou utlumeny.



Obr. 1- Spektrum zvukového záznamu Mp3

Proto se výsledky velikosti komprese liší v závislosti na šířce frekvenčního spektra a velikosti vzorkovacího kmitočtu. Je tedy nutné pro měření vybrat rozdílné zvukové záznamy se stejnou vzorkovací frekvencí, aby bylo možno pozorovat rozdílnou velikost komprese. Například čtený text, jazz, operu a elektronickou hudbu. Pro zpřesnění měření jsou vybraná celá alba, aby byly co největší časové rozdíly převodu. Alba jsou z originálního CD převedena na jeden souvislý wav.

Vybraný zvukový záznamy

| Název | typ | čas záznamu | vzorkovací frekvence | rozsah úrovní | kanál | kódování | formát záznamu |
|------------------|--------------|-------------|----------------------|---------------|--------|----------|----------------|
| Cimrman | slovo | 74:46 | 44,1 kHz | 16-bit | stereo | PCM | WAV |
| Armstrong | jazz | 68:38 | 44,1 kHz | 16-bit | stereo | PCM | WAV |
| Vivaldi | opera | 40:06 | 44,1 kHz | 16-bit | stereo | PCM | WAV |
| 666 | elektronická | 69:53 | 44,1 kHz | 16-bit | stereo | PCM | WAV |

Tab. 2 - Vybrané zvukové záznamy a jejich data

Tato alba byla otestována v programu Audiochecker 1.2, zdali se jedná o bezeztrátovou kompresi. S následujícími výsledky:

666.ape

Conclusion: This track looks like CDDA with probability 100%

Hello Dolly - jazz.ape

Conclusion: This track looks like CDDA with probability 100%

Vivaldi.ape

Conclusion: This track looks like CDDA with probability 100%

Cimrman - slovo.ape

Conclusion: This track looks like CDDA with probability 100%

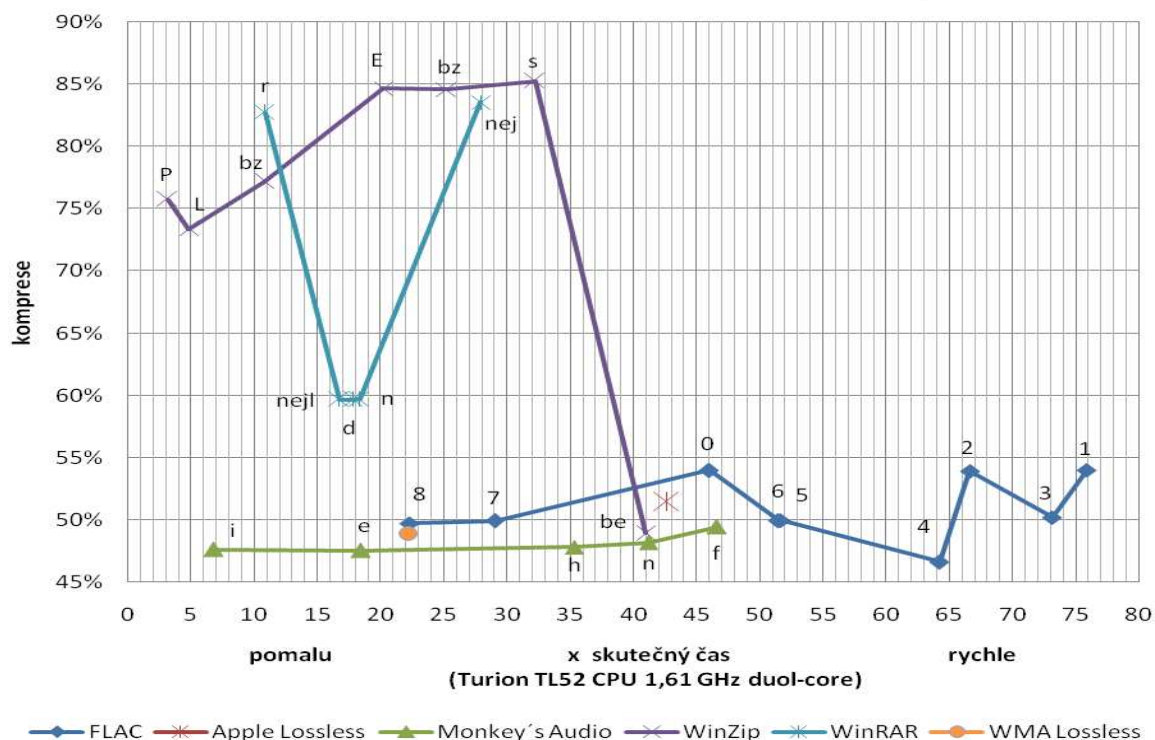
Zde je zobrazeno v procentech, s jakou pravděpodobností se jedná o daný typ záznamu. CDDA (Compact Disc Digital Audio systém) definuje standard pro zvukové CD.

3.5. Výsledky měření

Naměřené hodnoty jsou obsaženy ve čtyřech tabulkách v příloze (č. 1 – 4). Každá tabulka obsahuje název a typ zvukového záznamu, možnosti nastavení programů pro měření, výsledný faktor komprese, přepočítanou rychlost komprese a dekomprese.

Hodnoty jsou zpracovány do následujících osmi grafů, kde z každé tabulky vychází dva grafy. Jeden obsahuje kompresi (encoding) a druhý dekompresi (decoding). Na ose x se nachází rychlost procesu, která je dána poměrem délka záznamu děleno časem převodu. Na ose y vynášíme účinnost komprese v procentech. Ta udává, na jakou velikost z původní velikosti je komprimovaná. Zkratky nastavení komprese jsou tučně zobrazeny v tab. 2.

Cimrman - mluvené slovo encoding



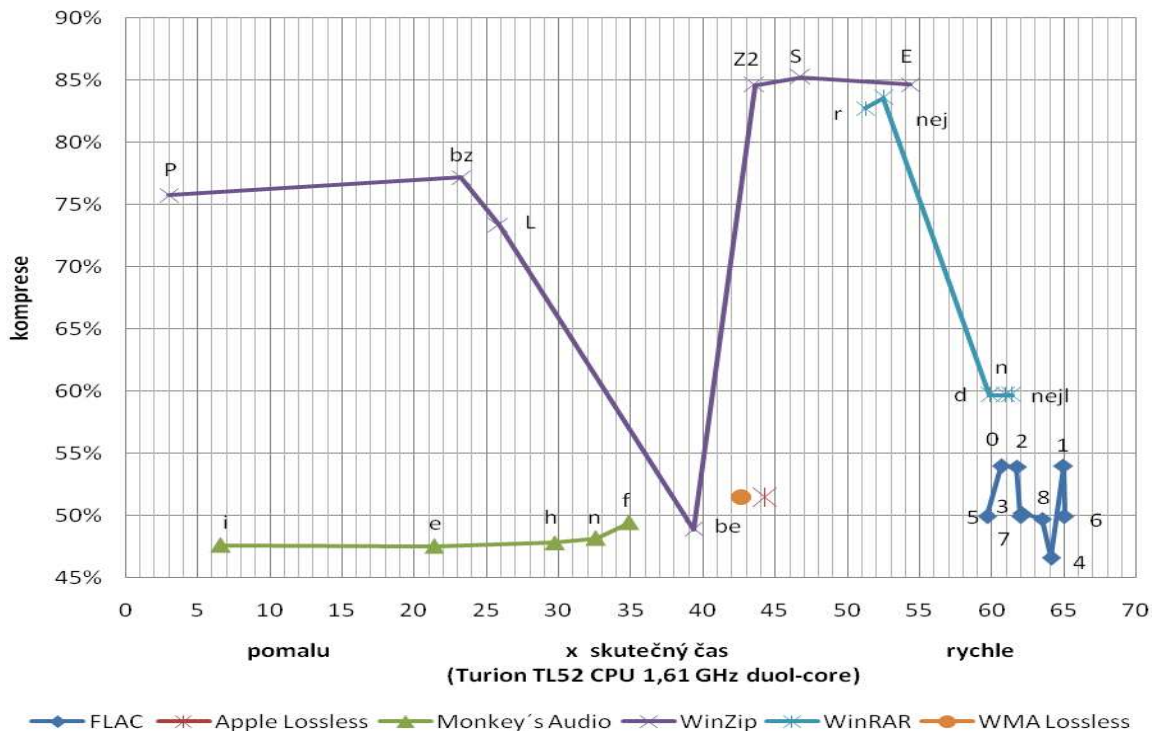
Obr. 2 - Cimrman – mluvené slovo encoding

Při prvním pohledu na Obr. 2. je vidět, že univerzální komprimační programy mají podstatě horší faktor komprese ve srovnání se zvukovými kompresními programy.

Konkurence schopný je WinZip s nastavením komprese na *best* (*be*). Mezi jednotlivými nastaveními komprese WinZipu jsou velké rozdíly ve velikosti komprese, která je v rozsahu od 46 % do 85 %. Nejlepších výsledků komprese dosahuje v průměru Monkey's Audio. U tohoto programu je rychlost komprimace a dekomprimace přibližně stejná. WMA má podobné vlastnosti jako FLAC s nastavením komprese 8, tato shoda je možná vidět u všech typů měřených zvukových záznamů, blízká je jak velikost komprese, tak rychlost komprimace. Zvukové kompresní programy dosahují u mluveného slova rozsahu komprese od 46 % do 54 %.

Z hlediska porovnání rychlosti komprese byl nejrychlejší FLAC, ten je dvakrát rychlejší než ostatní porovnávané programy. Apple je dvakrát rychlejší než konkurenční WMA.

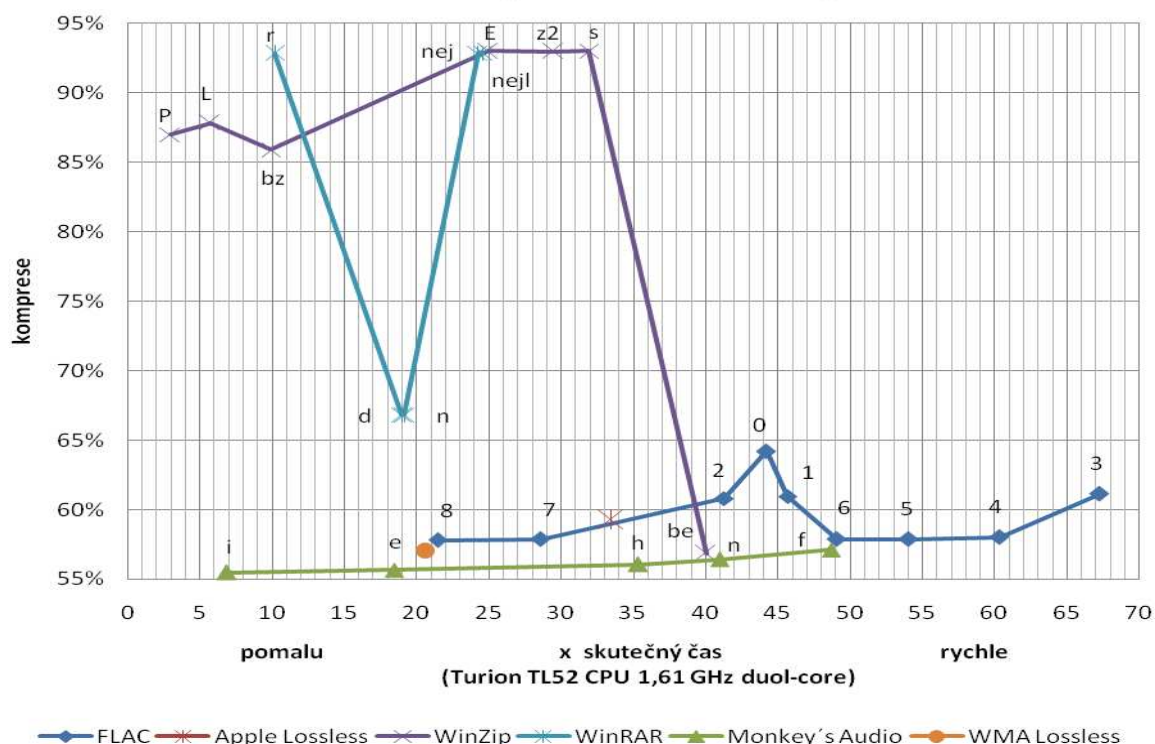
Cimrman - mluvené slovo decoding



Obr. 3 - Cimrman – mluvené slovo decoding

Při dekompresi je zřejmé z grafu, že kodek FLAC je opět nejrychlejší. Zároveň při všech nastaveních komprese má nepatrné rozdíly v rychlosti. Je evidentní, že proces dekomprese je pro všechna nastavení podobný. Při porovnání grafů kódování a dekódování je vidět, že nastavení, pro které je komprese nejrychlejší, neznamená, že bude i rychlá při dekompresi. Monkey's audio má pořadí nastavení stejné pro kompresi i dekompresi. Komprese se během měření pro jednotlivé zvukové záznamy pohybovala od 40 % do 60 %.

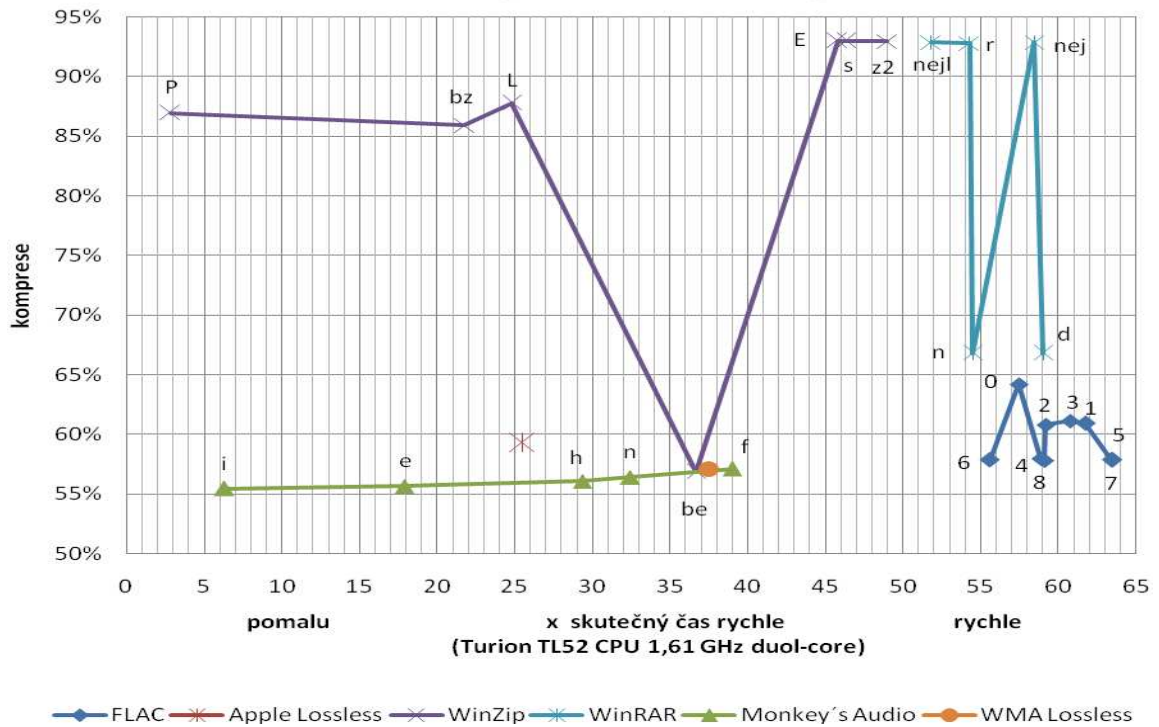
Armstrong - Jazz encoding



Obr. 4 - Armstrong – Jazz encoding

Při srovnání Obr. 2 s Obr. 4 si můžeme povšimnout hlavního rozdílu ve velikosti komprese, ta je u jazzu oproti mluvenému slovu přibližně o 10 % horší, což je nejspíše dáno složením zvukového záznamu a spektrem. Největší změna komprese nastala u WinRARu s nastavením na nejlepší. Ta se z předpokládaných 67 % (dle Obr. 2 a Obr. 8 se komprese pohybuje na úrovni s nastavením d a n) zhoršila na 92 % (Obr. 4). U mluveného slova má WinRAR s nastavením komprese *nejlepší*, *dobrá* a *normál* stejně velkou kompresi, na rozdíl od jazzu, kde pro nastavení komprese *dobrá* a *normál* je téměř stejná hodnota komprese a rychlosti. Komprese s nastavením *nejlepší* se přesunula na stejnou úroveň komprese a času jako je *nejrychlejší*. Tato charakteristika komprese je podobná s charakteristikou opery v Obr. 6.

Armstrong - Jazz decoding



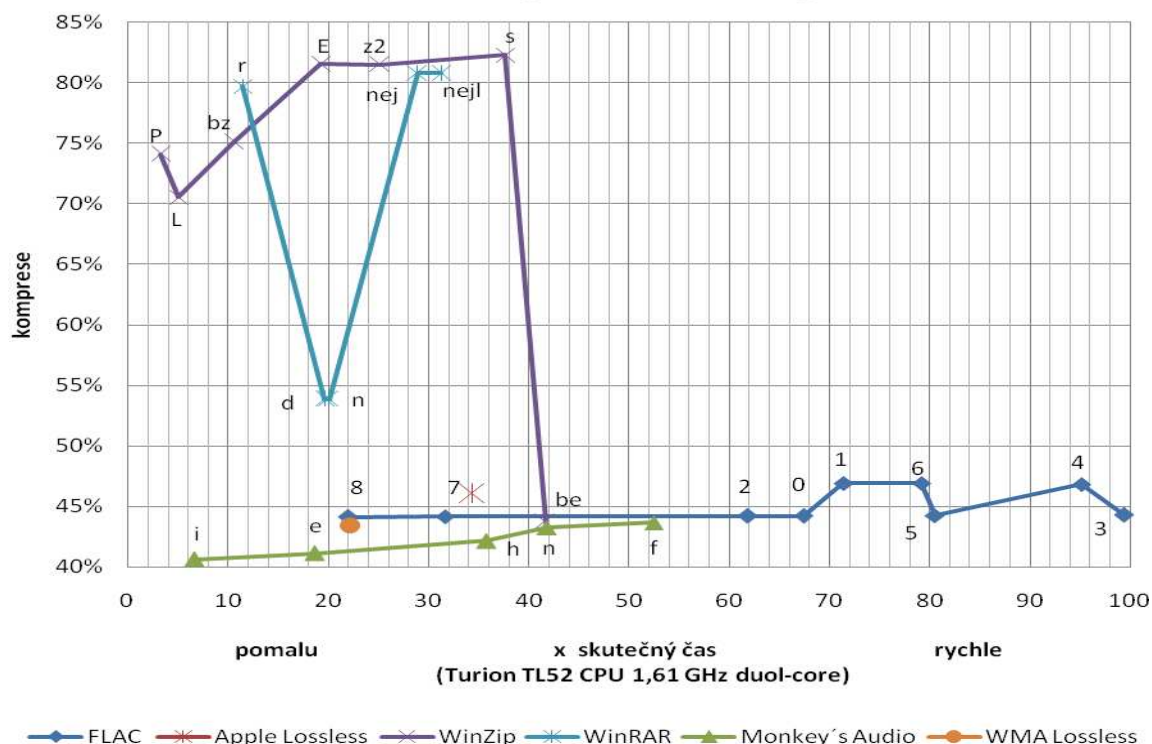
Obr. 5 - Armstrong - Jazz decoding

U dekomprese jazzu jako jediného je možno pozorovat rozdíl v rychlosti dekomprese, kde WMA je rychlejší než Apple. Při dekompresi ostatních zvukových záznamů je čas převodu WMA a Apple obrácený.

Při pohledu na všechny grafy můžeme pozorovat, že u FLACu není přímá závislost mezi nastavením stupně komprese, časem a velikostí komprese. Během komprese dosahuje rychlost FLACu od 21 do 67 (Obr. 4) zatím co u dekomprese je rozsah rychlosti od 55 do 58 (Obr. 5), z toho vyplývá, že mezi jednotlivými nastaveními komprese nepozorujeme vzájemnou závislost.

U komprese FLACu je nejpomalejší nastavení komprese 8, poté následuje 7, ostatní jsou různé. Dále pozorujeme posloupnost nastavení komprese 6, 5 a 4. V Obr. 2 vidíme, že FLAC s nastavením komprese 4 má větší kompresi než ostatní nastavení.

Vivaldi - opera encoding

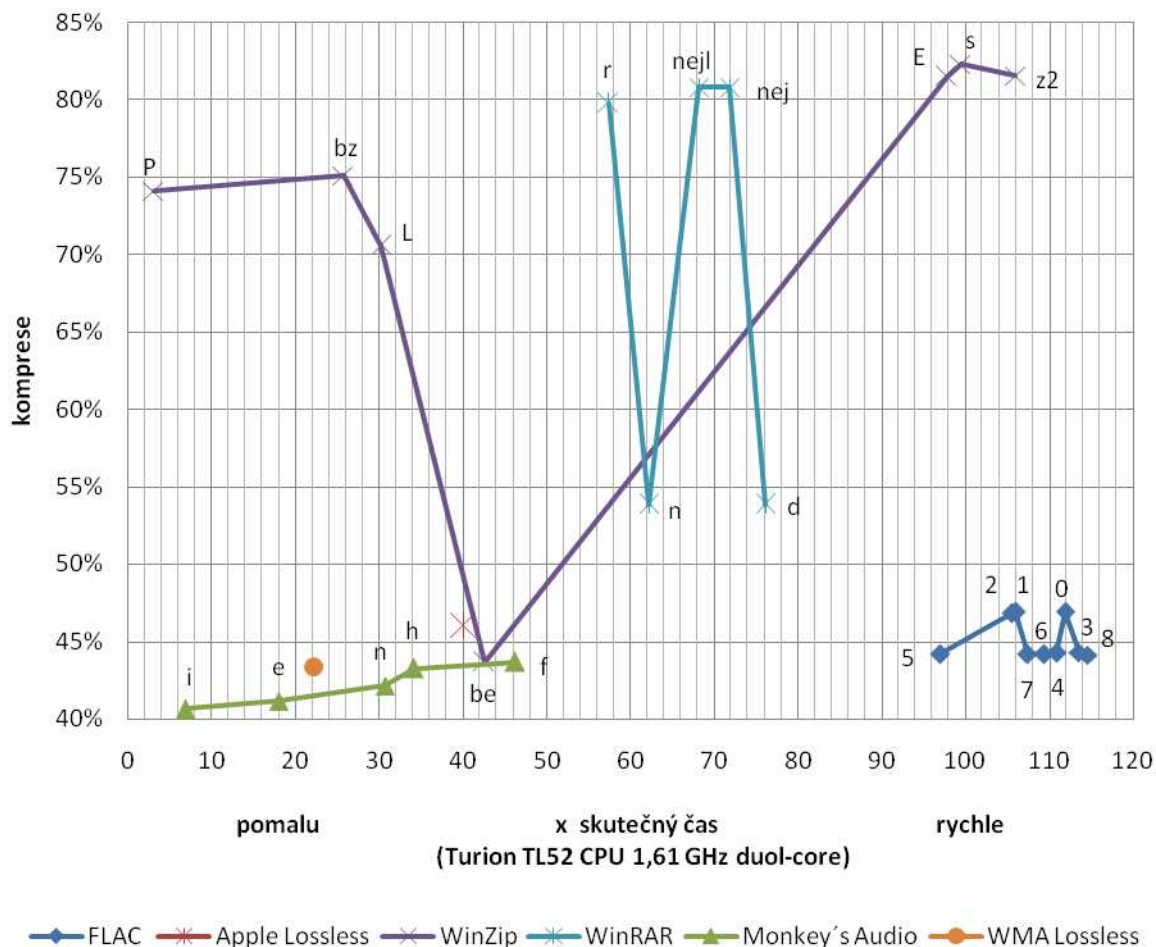


Obr. 6 - Vivaldi - opera encoding

U komprese zvukového záznamu opery byla dosažena největší komprese ze všech zkoumaných záznamů. Tu dosáhl Monkey's Audio při nastavení *insane* (i).

Při porovnání WinZipu při kompresi jednotlivých zvukových záznamů má křivka u opery (Obr. 6) podobný průběh jako u mluvené slovo (Obr. 2). Naproti tomu u jazzu (Obr. 4) vidíme že LZMA (L) má horší kompresi než nastavení *bzip2* (bz) či *PPMd* (P). V případě komprese elektronické hudby (Obr. 8) má WinZip s nastavením LZMA, PPMd a *bzip2* podobnou velikost komprese.

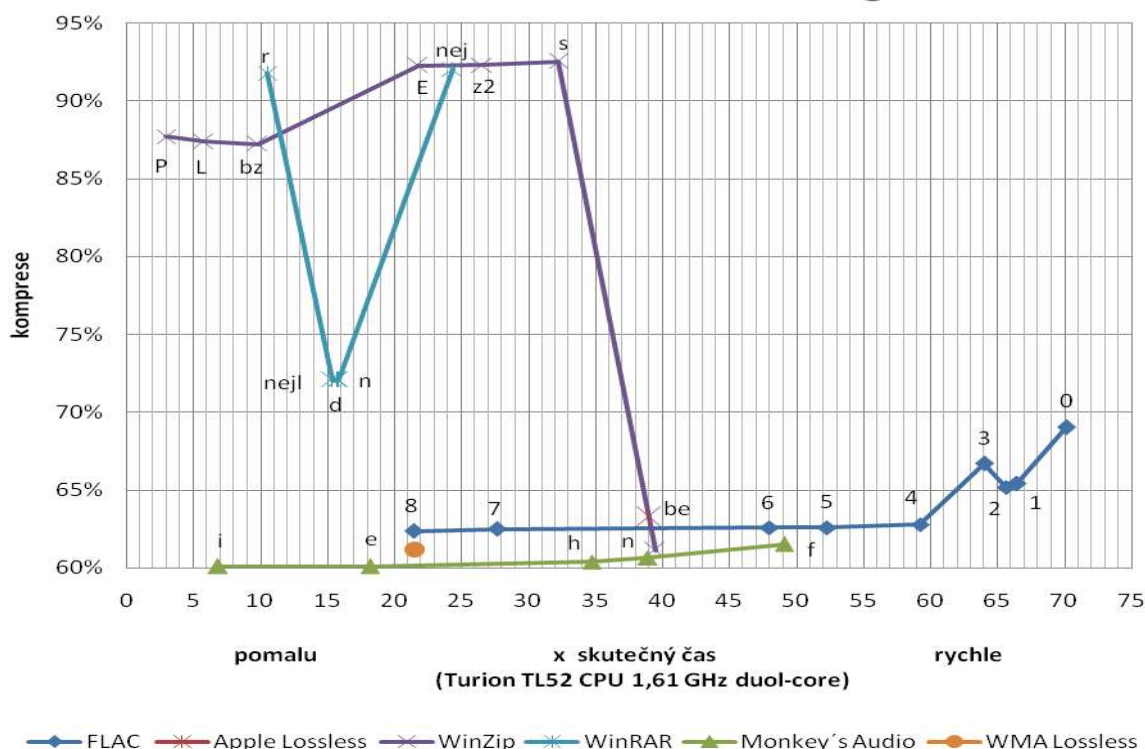
Vivaldi - opera decoding



Obr. 7 - Vivaldi - opera decoding

WinZip obsahuje rozdíly v rychlosti komprese a dekomprese, zatímco nastavení pro kompresi *best* a *PPMd* má přibližně totožnou rychlost jak pro kompresi tak dekompresi. Ostatní nastavení jsou pomalejší při kompresi a rychlejší při dekompresi. Nejrychlejší pro komprimaci je nastavení *best* a pro dekomprimaci to je *Zip 2 (z2)*.

666 - elektronická encoding



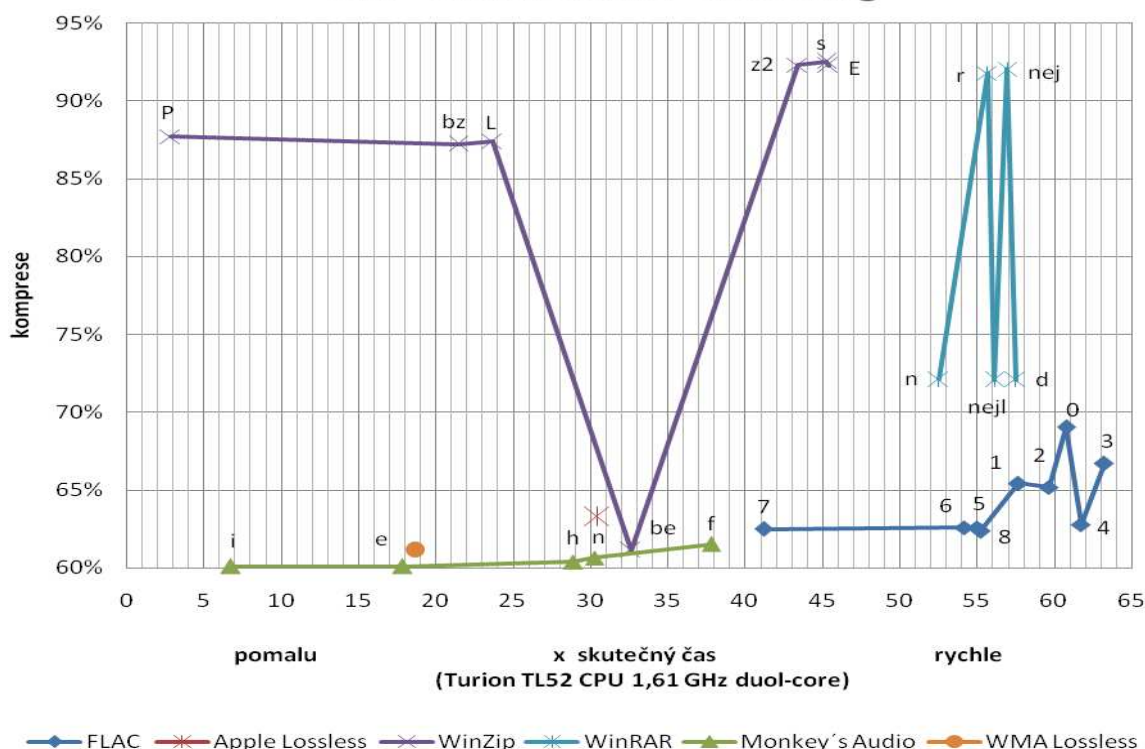
Obr. 8 - 666 - elektronická encoding

Při kompresi elektronické hudby se ukazuje, že je nejnáročnější na kompresi, což dokazují naměřené hodnoty pro různá kompresní nastavení.

FLAC v Obr. 8 má část pro nastavení komprese 4 až 8 téměř lineární, což znamená, že je efektivnější používat nastavení 4, které je třikrát rychlejší než nastavení 8. V Obr. 6 (opera) je možno vidět, že úroveň komprese je podobná pro nastavení 8, 7, 2, 0, 5 a 3. Oproti tomu rozdíl v čase je maximální u FLACu, kde nastavení 3 je pětkrát rychlejší než 8. U jazzu (Obr. 4) se nachází podobná komprese pro nastavení komprese 8, 7, 6, 5 a 4. V případě mluveného slova (Obr. 2) jsou to 8, 7, 6, 5 a 3.

Zajímavé jsou výsledky FLACu při dekompresi. Například v Obr. 9, je pořadí rychlosti pro různé nastavení komprese naprosto jiné pro každý zvukový záznam. Pozorujeme, že například nastavení 8 má rychlost komprese 21 a při dekompresi 55, z čehož vyplývá, že komprese je pomalejší než dekomprese. Naproti tomu nastavení 0 má rychlost komprese 70 a při dekompresi 61. Jednotlivá nastavení určují, zda čas komprese bude trvat déle či méně než dekomprese.

666 - elektronická decoding



Obr. 9 - 666 - elektronická decoding

3.6. Zhodnocení výsledků

Obr. 2 – 9 zachycují výsledky komprimace jednotlivých komprimačních programů a jejich nastavení. Níže je uvedena tabulka, v níž je přehled minimálních a maximálních naměřených hodnot.

| | komprese | | encoding | decoding | |
|-----------|----------|--------------------|----------|----------|-------------------|
| Cimrman | 85,22% | minimální komprese | 75,91 | 65,01 | největší rychlost |
| | 46,62% | maximální komprese | 3,16 | 2,98 | nejmenší rychlost |
| Armstrong | 93,01% | minimální komprese | 67,29 | 63,45 | největší rychlost |
| | 55,47% | maximální komprese | 2,97 | 2,84 | nejmenší rychlost |
| Vivaldi | 82,30% | minimální komprese | 99,42 | 114,57 | největší rychlost |
| | 40,64% | maximální komprese | 3,28 | 3,10 | nejmenší rychlost |
| 666 | 92,53% | minimální komprese | 70,12 | 63,24 | největší rychlost |
| | 60,09% | maximální komprese | 2,95 | 2,84 | nejmenší rychlost |
| Celkově | 93,01% | minimální komprese | 99,42 | 114,57 | největší rychlost |
| | 40,64% | maximální komprese | 2,95 | 2,84 | nejmenší rychlost |

Tab. 3 - Přehled hodnot komprese a časů

Pokud vybereme pouze programy určené pro zvukovou kompresi, je zde vidět, že porovnávané bezztrátové zvukové komprimační programy dosahují komprimace v rozsahu od 40,64 do 69,04 %. Nejlepšího výsledku 40,64 % dosáhl Monkey's Audio s nastavením komprese *insane*. Nevýhodou je malá rychlost, proto je lepší volit menší kompresi pro dosažení větší rychlosti. To vyplývá z faktu, že mezi největší a nejmenší kompresí je pouze minimální rozdíl, zatímco v čase několika-násobný.

| | komprese | | encoding | decoding | |
|-----------|----------|--------------------|----------|----------|-------------------|
| Cimrman | 54,01% | minimální komprese | 75,91 | 65,01 | největší rychlost |
| | 46,62% | maximální komprese | 6,84 | 6,55 | nejmenší rychlost |
| Armstrong | 64,20% | minimální komprese | 67,29 | 63,45 | největší rychlost |
| | 55,47% | maximální komprese | 6,84 | 6,30 | nejmenší rychlost |
| Vivaldi | 46,93% | minimální komprese | 99,42 | 114,57 | největší rychlost |
| | 40,64% | maximální komprese | 6,65 | 6,86 | nejmenší rychlost |
| 666 | 69,04% | minimální komprese | 70,12 | 63,24 | největší rychlost |
| | 60,09% | maximální komprese | 2,95 | 2,84 | nejmenší rychlost |
| max | 69,04% | minimální komprese | 99,42 | 114,57 | největší rychlost |
| min | 40,64% | maximální komprese | 2,95 | 2,84 | nejmenší rychlost |

Tab. 4 - Přehled minimálních a maximálních hodnot pouze zvukových programů

Jako nejefektivnější se jeví program FLAC, ten má sice o něco menší kompresi, zato je oproti ostatním komprimačním programům rychlejší. Také záleží, jaké nastavení komprese použijeme. Z univerzálních komprimačních programů je nejlepší WinZip s nastavením *best*. Tato komprese je srovnatelná s Monkey's Audio s nastavením *normal*, který má přibližně o 0,5 % lepší kompresi. WinZip má nejrychlejší komprimaci z univerzálních komprimačních programů, u dekomprese je pouze průměrný.

Komprese a rychlost je ovlivněna složením zvukového záznamu. Z těchto důvodů se provádělo měření ve vybraných stylech zvuku. Hlavní rozdíly jsou vidět v kompresi, kde se nejnáročnější na kompresi jeví elektronická hudba s nejlepší kompresí 60,09 %. Zatímco na druhé straně nejlepší komprese byla dosažena 40,64 % u opery. Při seřazení od nejlepší komprese po nejhorší je pořadí: opera, mluvené slovo, jazz a elektro.

4. OVĚŘOVÁNÍ BEZEZTRÁTOVÉ KOMPRESI

Kvalitu bezztrátového zvukového souboru lze ověřit. Například při stažení zvukového souboru z internetu či nagrabovaného z CD. Chceme-li mít jistotu, že dané soubory jsou bezztrátově komprimované z originálů nikoliv pouze rekomprimované již z provedené ztrátové komprimace audio souboru. Pro ověřování kvality zvukových záznamů lze použít ruční nebo automatickou metodu.

4.1. Ruční porovnání

Tato metoda vyžaduje mít originály ověřovaných zvukových souborů. Jedná se o náročnější postup. Kontrolovaný bezztrátový soubor musíme převést na WAVy. Poté můžeme porovnat MD5 (Message-Digest algoritmus) neboli *hash* (kontrolní součet) originálu a zkoumaného souboru. Jakákoli změna v řetězci zvukového souboru vyvolá změnu v MD5. Pokud se MD5 originálu a ověřovaného souboru shodují, jde o bezztrátovou kompresi.

4.2. Automatické ověřování

Pro automatickou kontrolu existuje několik programů, např. Foobar2000 přehrávač, Tau Analyzer, auCDTect verze 0.8 a Audiochecker. Výhoda této metody spočívá v tom, že ověření, zda se jedná o bezztrátovou kompresi, nepotřebuje originál.

Při kontrole se užívá metoda auCDTect, která analyzuje a matematicky vyhodnotí. Je vysoce přesná, pravděpodobnost bezztrátového souboru je vyjádřena v procentech. Nespornou výhodnou programem Audiochecker je vstup bezztrátových formátů APE, FLAC, SHN, WAV a LPAC. Vzhledem k tomu není nutno převádět ověřovaný zvukový soubor.

Tau Analyzaer využívá matematický postup a zobrazí frekvenční spektrum.

5. VYTVOŘENÍ JEDNODUCHÉHO BEZEZTRÁTOVÉHO KODÉRU A DEKODÉRU

Na začátku je nutné si stanovit výchozí požadavky pro tento úkol. Vstupní zvukový soubor bude ve formátu WAV. Následuje výběr programovacího prostředí, ve kterém bude vytvořen komprimační a dekomprimační program. Za tímto účelem jsem si zvolil programovací prostředí Microsoft Visual Studio 2005 jazyk C# z důvodů rostoucí popularity tohoto jazyka a rozšíření svých znalostí o tomto jazyce. Poslední a rozhodující je výběr algoritmu pro kompresi zvukových záznamů. Při výběru je rozhodující složitost a jeho účinnost. Z těchto aspektů jsem zvolil Run Length Encoding.

5.1. Run-length encoding

Kóduje shodné hodnoty jdoucích bajtů v řadě za sebou do dvojic skládajících se z dané hodnoty bajtů a počtu opakování. Účinnost komprese je silně závislá na charakteru vstupních dat, která musí obsahovat delší sekvence stejných hodnot, v opačném případě účinnost komprese výrazně klesá nebo dojde dokonce ke zvětšení objemu dat. RLE existuje v různém provedení. Pro ukázkou je níže uveden příklad na řadě písmen.

Příklad

Vstupní data před kódováním RLE:

```
aaaaaaaaaaaaadddddfffffggggggew
```

Výsledek kódování RLE:

```
a13d6f7g6e1w1
```

5.2. Řešení algoritmu

Problém nastává při rozboru daných wav souborů. V případě, kdy se daný bajt s následujícím neshoduje, je daný bajt zapsán jako dva bajty. Proto se může lehce stát, že místo abychom daný soubor zkomprimovali, dojde ke zvětšení. Je tedy nezbytné, komprimovat pouze data, která se opakují. Dva bajty po sobě také nemají smysl komprimovat, protože by byly zapsány jako dva bajty. Z toho plyne, že bajty,

které se neopakují, budou zapsány stejně a kódovány pouze budou znaky tři a více. Princip je vysvětlen na následujícím příkladu.

Příklad

Vstupní data před kódováním RLE:

aaaaaaaaaaaaadddddfffffggggggew

Výsledek kódování RLE:

Aaa10ddd3fff5ggg3ew

Tři stejné bajty po sobě jsou z důvodů dekomprese, aby bylo možno rozpoznat, jaké bajty byly zakódovány a jaké ne. Tato metoda je použita z protokolu MNP třídy 5 (Microcom Networking Protocol), který je využíván výrobci síťových modemů. Tento protokol využívá kombinaci dvou komprimačních metod – metody proudové komprimace RLE a adaptivní frekvenční metody [1].

5.3. Provedení

Program má šest tlačítek, kde pro každou operaci jsou tři tlačítka otevřít, uložit a start. První řada tlačítek je určena pro kompresi a spodní pak pro dekompresi. Pod nimi je umístěn vizuální ukazatel aktuálního stavu procesu. Pro měření času je ve spodní části umístěn ukazatel, který ukáže konečný čas operace. Wav soubor obsahuje na začátku RIFF [37], jedná se o hlavičku, ve které jsou zakomponovány informace a záznam o skladbě (vzorkovací frekvence, počet kanálů, kódování, počet úrovní vzorku atd.). Pokud je RIFF obsažen v daném souboru, zobrazí se hodnoty v pravé části. Grafické provedení je zobrazeno na obrázku níže. Podrobnější popis funkce je možno vidět na vývojovém diagramu v příloze č. 5.



Obr. 10 - Uživatelské rozhraní v PC

5.4. Ověření funkce

Pro ověření funkce kompresního a dekompresního programu jsem použil 8bitovou nahrávku, která je obsažena na příloženém CD. Velikost zvukového záznamu před kompresí je 534 579 bajtů. Na Obr. 11 je možno vidět, jak se shodné bity opakují v návaznosti. To je hlavní předpoklad pro kompresi touto metodou RLE.

```

7D 7F 80 82 84 85 84 83 80 7E 7D 7E 7F 81 82 83
83 82 81 81 80 7E 7D 7C 7B 7B 7C 7E 7F 80 81 80
80 7F 7F 7F 80 82 83 84 84 83 81 7F 7D 7C 7C 7C
7E 7F 80 81 81 80 7F 7E 7E 7F 81 82 83 84 83 82
81 80 7E 7D 7D 7D 7D 7F 80 80 80 7F 7E 7E 7E 7F
80 82 83 83 82 81 7F 7D 7B 79 79 7A 7C 7E 80 81
82 82 83 83 83 84 85 86 86 86 85 84 82 80 7E 7C
7C 7C 7D 7E 80 81 82 82 80 7E 7C 7B 7B 7C 7D
7D 7E 7D 7C 7B 7A 7A 7B 7C 7F 82 84 86 87 86 85
83 82 80 7F 7F 7E 7E 7E 7D 7D 7C 7C 7D 7D 7E
7F 81 81 82 82 81 80 7F 7E 7E 7F 80 80 81 80 80
7F 7E 7E 7E 80 81 83 84 85 84 83 81 7F 7E 7D 7D
7D 7D 7D 7E 7E 7F 80 80 80 80 80 7F 7F 7F 7F 7F
80 80 81 81 81 81 81 81 81 81 81 81 80 80 80 80
81 81 82 81 81 80 7F 7D 7C 7C 7C 7C 7C 7D 7E 7E
7F 7F 7F 7E 7E 7D 7D 7D 7E 7E 7E 7D 7D 7D 7D
7E 7F 80 81 81 82 83 83 82 82 80 7F 7E 7D 7D 7D
7E 7F 7F 80 81 82 83 84 85 85 84 84 82 81 80 7F
7F 7F 80 81 81 81 81 81 81 81 80 7F 7E 7E 7E 7E
7F 7F 80 81 82 84 84 84 83 82 80 7F 7D 7D 7D 7D

```

Obr. 11 – Ukázka výpisu 8-bit původní zvukový záznam

Po provedení komprese dostaneme záznam s příponou .R66 o velikosti 467 928 bajtů. Z těchto hodnot vypočteme kompresi.

$$\frac{467928}{534579} \cdot 100 = 87,5\%$$

V Obr. 12 jsou zobrazeny již zakódované bajty vyskytující se maximálně tři stejné bajty v řadě a na čtvrtém místě se nachází počet následujících stejných bajtů.

```

6D 6C 6B 6A 69 68 69 6B:6E 72 75 79 7D 81 84 87
89 8B 8C 8C 8C 00 8B 89:87 84 81 7F 7C 7A 77 75
73 72 71 72 74 77 7A 7D:80 82 83 84 84 83 83 82
82 82 00 84 87 8A 8D 8F:90 90 90 00 8E 8D 8B 8A
89 88 87 86 83 80 7B 76:72 70 70 71 73 75 77 78
7A 7B 7C 7E 7F 81 83 85:87 88 88 87 84 81 7D 79
76 74 73 73 74 75 77 79:7A 7C 7D 7E 7F 81 83 85
86 87 86 85 84 82 82 82:00 83 85 86 86 86 00 85
84 83 82 82 82 00 83 84:85 85 84 82 7F 7D 7B 7A
7A 7A 00 7B 7C 7D 7E 7F:80 81 82 82 82 00 83 83
83 02 82 81 80 7F 7E 7D:7D 7D 00 7E 80 82 85 87
88 88 87 87 86 84 83 81:7E 7B 76 71 6A 62 5B 53
4C 48 48 4C 55 63 71 7F:89 8D 8D 89 84 81 82 89
95 A3 B0 B8 B8 B1 A3 92:82 77 71 72 76 7D 83 87
88 86 83 7F 7B 7A 7B 7F:84 8A 90 94 95 93 8E 87
81 7C 7B 7F 86 8F 97 9C:9C 98 8E 83 77 6C 64 5F
5D 5E 60 62 65 67 69 6B:6E 70 73 75 77 79 7B 7E
82 87 8E 95 9B A0 A2 A2:9F 9A 94 8E 89 85 81 7E
7B 76 6F 67 5F 56 51 4E:51 56 5F 69 72 7A 7E 80
80 80 00 82 87 90 9A A5:AE B4 B5 B1 A8 9C 90 84
7B 75 72 71 71 70 6D 6A:65 61 5F 60 64 6B 75 80
8A 91 95 96 95 93 92 92:94 97 9B A0 A2 A2 9F 98
8F 85 7C 74 6F 6C 6A 68:67 65 63 61 61 62 66 6B

```

Obr. 12 – Ukázka výpisu 8-bit po komprimaci RLE

Veškeré použité prostředky pro ověření funkce jsou obsaženy na příloženém CD. Samotný program včetně zdrojových kódů, zvukový záznam ve formátu wav a komprimovaný s příponou .R66.

Výsledek je značně ovlivněn kvalitou záznamu. Nevhodné je pro více-kanálové či více jak 8 bitové záznamy.

ZÁVĚR

Úvodní část práce se věnuje rozdělení a stručnému popisu kompresních algoritmů. Následuje popis vybraných kompresních programů, jejich možnosti nastavení a vlastností.

Praktická část byla zaměřena na porovnání vybraných bezztrátových kompresních zvukových a univerzálních kompresních programů. Z měření, které jsem provedl, je vidět, že mezi univerzálními komprimačními programy a programy určenými pouze pro kompresi zvuku je značný rozdíl. Při použití univerzálního komprimačního programu pro kompresi zvukového záznamu je nevhodnější WinZip s nastavením komprese *best*. Při pohledu na grafy je vidět, že tato komprese i rychlost je průměrná mezi zvukovými kompresními programy. Z mého pohledu se jeví jako ideální bezztrátové programy Monkey's Audio a FLAC. Záleží, zdali máme hlavní požadavky na rychlost či kompresi. FLAC má nespornou výhodu v rychlosti dekomprese, která je rychlá pro všechny typy nastavení komprese.

V závěru práce bylo úkolem vytvoření bezztrátového kodéru a dekodéru. Při vytváření vlastního komprimačního programu jsem měl možnost poznat různé problémy spojené s kompresí, jako je uspořádání datového souboru a jeho hlavičky, rozeznávání zakódované a nekódované části. Měření provedené na vytvořeném kompresním programu jsou adekvátní k dané problematice a také k možnostem použitého algoritmu RLE.

POUŽITÁ LITERATURA

- [1] ČAPEK, Jan, FABIAN, Peter. *Komprimace dat: principy a praxe*. 1. vyd. Praha: Computer Press, 2000. Internet. ISBN 80-7226-231-9. Prezentace a komprimace zvukových souborů, s. 39-56.
- [2] *Hudba z počítače: Dejte přednost kvalitě* [online]. 2006 [cit. 2008-03-25]. Dostupný z WWW: <<http://www.zive.cz/default.aspx?article=131549>>.
- [3] *Performance comparison of lossless audio compressors* [online]. 2005-02-07 [cit. 2008-03-25]. Dostupný z WWW: <<http://members.home.nl/w.speek/comparison.htm>>.
- [4] HANS. *Compression and speed of lossless audio formats* [online]. 2006 [cit. 2008-03-25]. Dostupný z WWW: <<http://web.inter.nl.net/users/hvdh/lossless/lossless.htm>>.
- [5] *Kodek* [online]. 26. 2. 2008 [cit. 2008-03-25]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Kodek>>.
- [6] TIŠNOVSKÝ, Pavel. *PCX prakticky: implementace komprimace RLE* [online]. 23. 11. 2006 [cit. 2008-03-25]. Dostupný z WWW: <<http://www.root.cz/clanky/pcx-prakticky-implementace-komprimace-rle/>>.
- [7] *ZIP* [online]. 2007, 26. 12. 2007 [cit. 2008-04-04]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/ZIP>>.
- [8] *Komprese dat* [online]. 2008, 2. 3. 2008 [cit. 2008-04-04]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Komprese_dat>.
- [9] *Bezeztrátová komprese* [online]. 2008, 27. 2. 2008 [cit. 2008-04-04]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Bezeztr%C3%A1tov%C3%A1_komprese>.
- [10] *Apple Lossless* [online]. 2008, 12. 3. 2008 [cit. 2008-04-04]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Apple_Lossless>.

- [11] Free Lossless Audio Codec [online]. 2008, 18. 2. 2008 [cit. 2008-02-03].
Dostupný z WWW: <http://cs.wikipedia.org/wiki/Free_Lossless_Audio_Codec>.
- [11] *Performance comparison of lossless audio compressors* [online]. 2005, 2005-02-07 [cit. 2008-03-04]. Dostupný z WWW:
<<http://members.home.nl/w.speek/comparison.htm>>.
- [12] *Burrows-Wheelerova transformace* [online]. 2007, 10. 9. 2007 [cit. 2008-04-06].
Dostupný z WWW: <http://cs.wikipedia.org/wiki/Burrows-Wheelerova_transformace>.
- [13] *Move-to-front transform* [online]. 2008, 19. 2. 2008 [cit. 2008-04-06]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Move-to-front_transform>.
- [14] *STAUDEK, Jan. Kompresi dat* [online]. 2008 [cit. 2008-04-05]. Dostupný z WWW:
<http://www.fi.muni.cz/usr/staudek/vyuka/filesys/02_kompresi_dat.ps.pdf>.
- [15] *Huffmanovo kódování* [online]. 2007, 22. 8. 2007 [cit. 2008-04-06]. Dostupný z WWW:
<http://cs.wikipedia.org/wiki/Huffmanovo_k%C3%B3dov%C3%A1n%C3%AD>.
- [16] *Range encoding* [online]. 2008, 25. 3. 2008 [cit. 2008-04-06]. ENG. Dostupný z WWW: <http://en.wikipedia.org/wiki/Range_coding>.
- [17] *Algoritmy komprese dat: Další metody bezztrátové* [online]. 2007, 24.4.2007 [cit. 2008-04-06]. Dostupný z WWW:
<<http://ksvi.mff.cuni.cz/~dvorak/vyuka/SWI072/dalsi.pdf>>.
- [18] *Prediction by Partial Matching* [online]. 2008, 11. 2. 2008 [cit. 2008-04-06]. ENG. Dostupný z WWW:
<http://en.wikipedia.org/wiki/Prediction_by_Partial_Matching>.
- [19] TROCH, Josef. *Algoritmy LZFG* [online]. 2000, 8.11.2000 [cit. 2008-04-06]. Dostupný z WWW: <<http://jt.wz.cz/vytvory/lzfg/lzfg.htm>>.
- [20] *Kompresní metoda LZ78* [online]. 1999 [cit. 2008-04-06]. Dostupný z WWW: <<http://vychodil.inf.upol.cz/courses/cs1pp/doc/lz78.pdf>>.

- [21] *Lempel-Ziv-Markov chain algorithm* [online]. 2008, 2. 4. 2008 [cit. 2008-04-06]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Lempel-Ziv-Markov_chain_algorithm>.
- [22] *Run-length encoding* [online]. 2008, 12. 2. 2008 [cit. 2008-04-07]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Run_length_encoding>.
- [23] *Komprimační algoritmy* [online]. 2008 [cit. 2008-04-07]. Dostupný z WWW: <<http://gimli.mysteria.cz/komprese/algoritmy.html>>.
- [24] *Windows Media Audio 9 Lossless* [online]. 2008, 22. 3. 2008 [cit. 2008-04-16]. Dostupný z WWW: <http://en.wikipedia.org/wiki/WMA_Lossless>.
- [25] *Golomb coding* [online]. 2008, 18. 1. 2008 [cit. 2008-05-01]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Golomb_coding>.
- [26] *Linear prediction* [online]. 2008, 22. 4. 2008 [cit. 2008-05-02]. Eng. Dostupný z WWW: <http://en.wikipedia.org/wiki/Linear_prediction>.
- [27] HAMMERTON, David. *ALAC* [online]. 2006, 30. 9. 2006 [cit. 2008-05-02]. Eng. Dostupný z WWW: <<http://craz.net/programs/itunes/alac.html>>.
- [28] MATHEW , T. Ashland. *Monkey's audio: A fast and powerful lossless audio compressor* [online]. c2004 [cit. 2008-05-09]. Eng. Dostupný z WWW: <<http://www.monkeysaudio.com/>>.
- [29] *Monkey's Audio* [online]. 2008, 14. 4. 2008 [cit. 2008-05-09]. Eng. Dostupný z WWW: <http://en.wikipedia.org/wiki/Monkey's_Audio>.
- [30] *Windows Media Audio Codecs* [online]. c2008 [cit. 2008-05-10]. Eng. Dostupný z WWW: <<http://www.microsoft.com/windows/windowsmedia/forpros/codecs/audio.aspx#WindowsMediaAudio9Lossless>>.
- [31] *ZIP* [online]. 2008, 7 May 2008 [cit. 2008-05-10]. Eng. Dostupný z WWW: <http://en.wikipedia.org/wiki/ZIP_%28file_format%29>.

- [32] *RAR* [online]. 2008, 8 May 2008 [cit. 2008-05-10]. Dostupný z WWW: <http://en.wikipedia.org/wiki/RAR_%28file_format%29>.
- [33] DR. JÜRGEN, Abel. *Weighed Frequency Count (WFC)* [online]. c2009, 2009 [cit. 2009-03-30]. Eng. Dostupný z WWW: <<http://www.data-compression.info/Algorithms/WFC/index.htm>>.
- [34] DR. JÜRGEN, Abel. *Distance Coding (DC)* [online]. c2009 [cit. 2009-03-30]. Eng. Dostupný z WWW: <<http://www.data-compression.info/Algorithms/DC/index.htm>>.
- [35] DR. JÜRGEN, Abel. *Inversion Frequencies (IF)* [online]. c2009 [cit. 2009-03-30]. Eng. Dostupný z WWW: <<http://www.data-compression.info/Algorithms/IF/index.htm>>.
- [36] MIROSLAV, Virius. *C# : Hotová řešení*. 1. vyd. Brno: Computer Press, 2006. 341 s. ISBN 80-251-1084-2.
- [37] SCOTT, Wilson. *WAVE PCM soundfile format* [online]. 2003, 2003 [cit. 2009-04-16]. Eng. Dostupný z WWW: <<file:///localhost/C:/Documents%20and%20Settings/Vaclav%20LE/Dokumenty/!škola/!bakalařka/program/Microsoft%20WAVE%20soundfile%20format.mht>>.

SEZNAM OBRÁZKŮ

| | |
|--|----|
| Obr. 1- Spektrum zvukového záznamu Mp3 | 23 |
| Obr. 2 - Cimrman – mluvené slovo encoding | 25 |
| Obr. 3 - Cimrman – mluvené slovo decoding | 26 |
| Obr. 4 - Armstrong – Jazz encoding | 27 |
| Obr. 5 - Armstrong - Jazz decoding | 28 |
| Obr. 6 - Vivaldi - opera encoding | 29 |
| Obr. 7 - Vivaldi - opera decoding | 30 |
| Obr. 8 - 666 - elektronická encoding | 31 |
| Obr. 9 - 666 - elektronická decoding | 32 |
| Obr. 10 - Uživatelské rozhraní v PC | 37 |
| Obr. 11 – Ukázka výpisu 8-bit původní zvukový záznam | 37 |
| Obr. 12 – Ukázka výpisu 8-bit po komprimaci RLE | 38 |

SEZNAM TABULEK

| | |
|---|----|
| Tab. 1 - Možnosti nastavení komprese | 22 |
| Tab. 2 - Vybrané zvukové záznamy a jejich data | 23 |
| Tab. 3 - Přehled hodnot komprese a časů | 32 |
| Tab. 4 - Přehled minimálních a maximálních hodnot pouze zvukových programů | 33 |

| Cimrman - mluvené slovo | | | x reálný čas | |
|-------------------------|------------------|----------|--------------|----------|
| program | nastavení | komprese | encoding | decoding |
| FLAC | 1 | 54,01% | 75,91 | 64,92 |
| | 3 | 50,20% | 73,18 | 62,05 |
| | 2 | 53,93% | 66,66 | 61,71 |
| | 4 | 46,62% | 64,27 | 64,09 |
| | 5 | 49,96% | 51,68 | 59,65 |
| | 6 | 49,95% | 51,50 | 65,01 |
| | 0 | 53,99% | 46,01 | 60,62 |
| | 7 | 49,93% | 29,04 | 61,96 |
| | 8 | 49,73% | 22,30 | 63,45 |
| Apple Lossless | | 51,46% | 42,68 | 44,20 |
| Monkey's Audio | fast | 49,46% | 46,64 | 34,86 |
| | normal | 48,19% | 41,22 | 32,56 |
| | high | 47,85% | 35,31 | 29,70 |
| | extra high | 47,56% | 18,45 | 21,33 |
| | insane | 47,63% | 6,84 | 6,55 |
| WMA Lossless | | 48,93% | 22,15 | 21,93 |
| WinZip | best | 48,92% | 41,01 | 39,35 |
| | SuperFast | 85,22% | 32,23 | 46,73 |
| | zip 2 | 84,58% | 25,23 | 43,55 |
| | Enhanced Deflate | 84,64% | 20,24 | 54,31 |
| | bzip2 | 77,18% | 10,81 | 23,21 |
| | LZMA | 73,37% | 4,83 | 25,81 |
| | PPMd | 75,73% | 3,16 | 2,98 |
| WinRAR | nejrychlejší | 83,55% | 27,93 | 52,47 |
| | normalní | 59,69% | 18,39 | 60,87 |
| | dobrá | 59,68% | 17,82 | 59,81 |
| | nejlepší | 59,67% | 16,76 | 61,37 |
| | rychlá | 82,72% | 10,89 | 51,21 |

| Armstrong - Jazz | | | x reálný čas | |
|-----------------------|------------------|----------|--------------|----------|
| program | nastavení | komprese | encoding | decoding |
| FLAC | 3 | 61,14% | 67,29 | 60,74 |
| | 4 | 58,00% | 60,38 | 58,83 |
| | 5 | 57,88% | 54,04 | 63,35 |
| | 6 | 57,88% | 49,08 | 55,57 |
| | 1 | 60,95% | 45,65 | 61,74 |
| | 0 | 64,20% | 44,23 | 57,43 |
| | 2 | 60,80% | 41,26 | 59,17 |
| | 7 | 57,86% | 28,62 | 63,45 |
| | 8 | 57,81% | 21,46 | 59,08 |
| Apple Lossless | | 59,28% | 33,45 | 25,48 |
| Monkey's Audio | fast | 57,11% | 48,69 | 38,97 |
| | normal | 56,43% | 40,95 | 32,43 |
| | high | 56,07% | 35,36 | 29,34 |
| | extra high | 55,67% | 18,45 | 17,88 |
| | insane | 55,47% | 6,84 | 6,30 |
| WMA Lossless | | 57,01% | 20,56 | 37,54 |
| WinZip | best | 56,85% | 40,06 | 36,67 |
| | SuperFast | 93,01% | 31,92 | 46,32 |
| | zip 2 | 92,98% | 29,46 | 48,97 |
| | Enhanced Deflate | 92,99% | 24,96 | 45,76 |
| | bzip2 | 85,91% | 9,93 | 21,77 |
| | LZMA | 87,80% | 5,72 | 24,81 |
| | PPMd | 86,97% | 2,97 | 2,84 |
| WinRAR | nejlepší | 92,87% | 24,57 | 51,80 |
| | nejrychlejší | 92,87% | 24,27 | 58,41 |
| | normální | 66,83% | 19,16 | 54,47 |
| | dobrá | 66,83% | 18,99 | 59,00 |
| | rychlá | 92,81% | 10,17 | 54,26 |

| Vivaldi - opera | | | x reálný čas | |
|-----------------------|------------------|----------|--------------|----------|
| program | nastavení | komprese | encoding | decoding |
| FLAC | 3 | 44,29% | 99,42 | 113,49 |
| | 2 | 46,84% | 95,10 | 105,53 |
| | 4 | 44,26% | 80,47 | 110,88 |
| | 1 | 46,93% | 79,14 | 105,99 |
| | 0 | 46,91% | 71,39 | 111,91 |
| | 5 | 44,20% | 67,39 | 97,02 |
| | 6 | 44,20% | 61,85 | 109,36 |
| | 7 | 44,16% | 31,66 | 107,41 |
| | 8 | 44,12% | 22,03 | 114,57 |
| Apple Lossless | | 46,02% | 34,37 | 40,10 |
| Monkey's Audio | fast | 43,69% | 52,49 | 46,19 |
| | normal | 43,27% | 41,83 | 34,17 |
| | high | 42,15% | 35,84 | 30,68 |
| | extra high | 41,14% | 18,72 | 17,99 |
| | insane | 40,64% | 6,65 | 6,86 |
| WMA Lossless | | 43,37% | 22,30 | 22,11 |
| WinZip | best | 43,69% | 41,70 | 42,58 |
| | SuperFast | 82,30% | 37,65 | 99,42 |
| | zip 2 | 81,50% | 25,17 | 97,80 |
| | Enhanced Deflate | 81,54% | 19,28 | 105,99 |
| | bzip2 | 75,11% | 10,56 | 25,73 |
| | LZMA | 70,58% | 5,10 | 30,23 |
| | PPMd | 74,10% | 3,28 | 3,10 |
| WinRAR | nejlepší | 80,79% | 31,25 | 68,16 |
| | nejrychlejší | 80,79% | 28,92 | 71,82 |
| | normální | 53,86% | 20,07 | 76,14 |
| | dobrá | 53,85% | 19,66 | 62,17 |
| | rychlá | 79,75% | 11,47 | 57,42 |

| 666 - elektronická | | | x reálný čas | |
|-----------------------|------------------|----------|--------------|----------|
| program | nastavení | komprese | encoding | decoding |
| FLAC | 0 | 69,04% | 70,12 | 60,77 |
| | 1 | 65,41% | 66,45 | 57,68 |
| | 2 | 65,17% | 65,62 | 59,64 |
| | 3 | 66,70% | 64,02 | 63,24 |
| | 4 | 62,79% | 59,22 | 61,75 |
| | 5 | 62,59% | 52,28 | 54,95 |
| | 6 | 62,57% | 47,97 | 54,17 |
| | 7 | 62,47% | 27,62 | 41,19 |
| | 8 | 62,37% | 21,43 | 55,24 |
| Apple Lossless | | 63,26% | 38,93 | 30,43 |
| Monkey's Audio | fast | 61,51% | 49,09 | 37,80 |
| | normal | 60,63% | 38,87 | 30,25 |
| | high | 60,39% | 34,70 | 28,91 |
| | extra high | 60,09% | 18,15 | 17,85 |
| | insane | 60,09% | 6,80 | 6,73 |
| WMA Lossless | | 61,11% | 21,55 | 18,65 |
| WinZip | best | 61,11% | 39,44 | 32,60 |
| | SuperFast | 92,53% | 32,23 | 45,28 |
| | zip 2 | 92,32% | 26,52 | 43,41 |
| | Enhanced Deflate | 92,28% | 21,77 | 45,38 |
| | bzip2 | 87,24% | 9,78 | 21,46 |
| | LZMA | 87,38% | 5,75 | 23,68 |
| | PPMd | 87,71% | 2,95 | 2,84 |
| WinRAR | nejrychlejší | 92,05% | 24,32 | 56,97 |
| | normální | 72,09% | 15,80 | 52,54 |
| | dobrá | 72,07% | 15,67 | 57,52 |
| | nejlepší | 72,06% | 15,34 | 56,13 |
| | rychlá | 91,79% | 10,50 | 55,68 |

Vývojový diagram vytvořeného programu.

