

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

POROVNÁNÍ HLASOVÝCH A AUDIO KODEKŮ

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. MICHAL LÚDIK

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## POROVNÁNÍ HLASOVÝCH A AUDIO KODEKŮ

COMPARISON OF VOICE AND AUDIO CODECS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MICHAL LÚDIK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IVAN MÍČA

BRNO 2012



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Diplomová práce

magisterský navazující studijní obor  
Telekomunikační a informační technika

**Student:** Bc. Michal Lúdík

**ID:** 106599

**Ročník:** 2

**Akademický rok:** 2011/2012

**NÁZEV TÉMATU:**

## Porovnání hlasových a audio kodeků

### POKYNY PRO VYPRACOVÁNÍ:

Porovnejte mezi sebou volně dostupné implementace audio kodeků z hlediska výpočetní náročnosti. Vyberte a implementujte vhodnou objektivní metodu hodnocení kvality dle ITU, přičemž vycházejte z referenční implementace. S pomocí této objektivní metody porovnejte také kvalitu výstupů dostupných kodeků.

### DOPORUČENÁ LITERATURA:

[1] AOKI, N., TAKAYA, K. MELP vocoder using lifting wavelet transform. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 1999. Vol., no., pp.392-395, 1999.

[2] WANG, S., SEKEY, A., GERSHO, A. An objective measure for predicting subjective quality of speech coders. IEEE Journal on selected areas in communications. June 1992. ISSN 0733-8716.

**Termín zadání:** 6.2.2012

**Termín odevzdání:** 24.5.2012

**Vedoucí práce:** Ing. Ivan Míča

**Konzultanti diplomové práce:**

**prof. Ing. Kamil Vrba, CSc.**

*Předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Táto práca sa zaoberá popisom vlastností ľudského sluchu, audio a rečových kodekov, popisom objektívneho hodnotenia ich kvality a ich praktickým porovnaním. Kapitola o audio kodekoch zahŕňa popis bezstratového kodeku FLAC a stratových kodekov MP3 a Ogg Vorbis. Z rečových kodekov je v práci uvedený prehľad lineárneho predikčného kódovania a kodekov G.729 a OPUS. Hodnotenie kvality obsahuje popis segmentového merania odstupe signál-šum a perceptuálnych meraní ako sú WSS a PESQ. Záverečná kapitola je tvorená popisom praktickej časti tejto práce, kde bola pri audio kodekoch porovnaná pamäťová náročnosť a nároky na výpočtové prostriedky a pri rečových kodekoch objektívna miera PESQ.

## **KLÚČOVÉ SLOVÁ**

bezstratova, celp, flac, g.729, huffmanovo kodovanie, kompresia, linearna predikcia, LPC, mp3, objektivne merania, ogg, opus, pesq, porovnanie, psychoakusticky model, segmentove snr, stratova, valgrind, wss

## **ABSTRACT**

This thesis deals with description of human hearing, audio and speech codecs, description of objective measure of quality and practical comparison of codecs. Chapter about audio codecs consists of description of lossless codec FLAC and lossy codecs MP3 and Ogg Vorbis. In chapter about speech codecs is description of linear predictive coding and G.729 and OPUS codecs. Evaluation of quality consists of description of segmental signal-to-noise ratio and perceptual evaluation of quality – WSS and PESQ. Last chapter deals with description of practical part of this thesis, that is comparison of memory and time consumption of audio codecs and perceptual evaluation of speech codecs quality.

## **KEYWORDS**

lossless, celp, flac, g.729, huffman coding, compression, linear prediction, LPC, mp3, objective quality, ogg, opus, pesq, comparison, psychoacoustic model, segmental snr, lossy, valgrind, wss

LÚDIK, Michal *Porovnaní hlasových a audio kodekú*: semestrálny projekt. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2011. 61 s. Vedúci práce bol Ing. Ivan Míča



## PREHLÁSENIE

Prehlasujem, že som svoj semestrálny projekt na tému „Porovnaní hlasových a audio kodekú“ vypracoval samostatne pod vedením vedúceho semestrálneho projektu, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedeného semestrálneho projektu ďalej prehlasujem, že v súvislosti s vytvorením tohoto semestrálneho projektu som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia § 152 trestného zákona č. 140/1961 Sb.

Brno .....

.....

(podpis autora)

## POĎAKOVANIE

Rád by som na tomto mieste poďakoval vedúcemu diplomovej práce pánovi Ing. Ivanovi Míčovi za jeho cenné rady počas vypracovávaní projektu, ktoré boli či už teoretického, alebo praktického charakteru.

Brno .....

.....

(podpis autora)

# OBSAH

Úvod	10
<b>1 Vlastnosti ľudského sluchu</b>	<b>12</b>
1.1 Psychoakustický model . . . . .	12
<b>2 Audio kodeky</b>	<b>14</b>
2.1 FLAC . . . . .	14
2.1.1 Bezstratová kompresia . . . . .	14
2.2 MPEG Layer III . . . . .	16
2.3 Ogg Vorbis . . . . .	17
<b>3 Rečové kodeky</b>	<b>19</b>
3.1 Linear predictive coding . . . . .	19
3.2 Lineárna predikcia . . . . .	22
3.3 Code Excited Linear Prediction . . . . .	24
3.4 G.729 . . . . .	24
3.5 OPUS . . . . .	25
3.5.1 SILK . . . . .	27
3.5.2 CELT . . . . .	27
<b>4 Hodnotenie kvality</b>	<b>28</b>
4.1 Segmentové merania SNR . . . . .	28
4.2 Perceptuálne orientované merania . . . . .	29
4.2.1 WSS meranie . . . . .	29
4.2.2 PESQ . . . . .	30
<b>5 Porovnanie zvukových kodekov</b>	<b>31</b>
5.1 Pamäťová a časová náročnosť audio kodekov . . . . .	33
5.1.1 Odchýlky merania . . . . .	36
5.2 PESQ . . . . .	39
<b>6 Záver</b>	<b>41</b>
Literatúra	43
Zoznam symbolov, veličín a skratiek	47
Zoznam príloh	49

<b>A Príloha 1</b>	<b>50</b>
A.1 Implementácia lineárnej predikcie . . . . .	50
<b>B Príloha 2</b>	<b>52</b>
B.1 Grafický prehľad pamätovej náročnosti . . . . .	52
<b>C Príloha 3</b>	<b>61</b>
C.1 Obsah CD . . . . .	61

# ZOZNAM OBRÁZKOV

1.1	Princíp perceptuálneho kodéru . . . . .	12
1.2	Krivky rovnakej hlasitosti a prah počutia[27] . . . . .	13
2.1	Príklad Huffmanovho kódovania . . . . .	15
2.2	Príklad LZ kódovania . . . . .	15
2.3	Kodér MPEG Layer 3 . . . . .	16
3.1	Bloková schéma LPC kodéru . . . . .	20
3.2	Bloková schéma produkcie reči . . . . .	20
3.3	Úsek zvukovej nahrávky reči . . . . .	21
3.4	Ukážka slova „skúška“ . . . . .	21
3.5	Bloková schéma CELP kodéru . . . . .	24
3.6	Kodér a dekodér . . . . .	26
4.1	Princíp PESQ . . . . .	30
5.1	Systém v stave idle . . . . .	33
5.2	Proces kódovania wav do mp3 . . . . .	33
5.3	Systém v záťaži . . . . .	34
5.4	Odchýlky merania kodeku flac . . . . .	36
5.5	Odchýlky merania kodeku ffmpeg . . . . .	37
5.6	Odchýlky merania kodeku lame . . . . .	37
5.7	Odchýlky merania kodeku vorbis . . . . .	38
5.8	Odchýlky merania kodeku OPUS . . . . .	38
5.9	Ukážka výstupu referenčnej implementácie PESQ . . . . .	39
B.1	FFMPEG 128 kbps . . . . .	53
B.2	FFMPEG 320 kbps . . . . .	54
B.3	LAME 32 kbps . . . . .	55
B.4	LAME 128 kbps . . . . .	56
B.5	LAME 320 kbps . . . . .	57
B.6	Vorbis 32-64 kbps . . . . .	58
B.7	Vorbis 112 kbps . . . . .	59
B.8	Vorbis 224 kbps . . . . .	60

# ZOZNAM TABULIEK

5.1	Časy [s] potrebné na prevod skladieb do formátov . . . . .	32
5.2	Priemerné časy pre flac . . . . .	34
5.3	Priemerné časy pre ffmpeg . . . . .	35
5.4	Priemerné časy pre lame . . . . .	35
5.5	Priemerné časy pre vorbis . . . . .	35
5.6	Priemerné časy pre OPUS . . . . .	36
5.7	Hodnoty MOS po prechode PESQ algoritmu . . . . .	40

# ÚVOD

Od prvopočiatkov vzniku elektronických zariadení vznikali prístroje produkujúce alebo zaznamenávajúce zvuk. Jeho uchovávanie je významné z viacerých dôvodov. Sú to napríklad vojenské účely, ako napr. dôkazové materiály, alebo osobné, pre zábavu. Zvuk môže byť v zariadeniach uchovaný analógovo (magnetické pásky) alebo digitálne (pevné disky, pamäťové karty a pod.).

Pre uchovanie zvukového signálu digitálnou formou sa používajú rôzne kodeky, rečové a zvukové, v závislosti na tom, aký zdroj kódujeme. Tieto sa ďalej rozdeľujú na stratové a bezstratové. Existuje ich veľké množstvo a na každý sa uplatňujú rôzne licenčné podmienky.

Medzi najpoužívanejšie audio kodeky patria bezstratový FLAC (Free Lossless Audio Codec), ktorý využíva huffmanovo alebo LZ kódovanie, alebo WAVE (Waveform Audio File Format), ktoré využíva pulzne kódovú moduláciu. Kodeky využívajúce stratové kódovanie sú OGG Vorbis alebo najpoužívanejší MP3 (MPEG Layer III). Dňa 1. 11. 2011 vyšla na internete správa, že spoločnosť Apple uvoľňuje svoj kodek ALAC (Apple Lossless Audio Codec)[7]. Tento sa však používa prevažne v zariadeniach Apple a nie je taký známy a rozšírený ako kodeky spomínané vyššie.

Medzi najznámejšie rečové kodeky patria napríklad G.711, či G.729. Využívajú sa prevažne v IP telefónii a líšia sa prenosovou rýchlosťou a použitou kompresnou metódou. Kým prenosová rýchlosť G.711 je 64 kbps, u G.729 je to menej než 8 kbps. Dátovému toku odpovedá aj príslušná kvalita.

Špeciálnym kodekom je kodek OPUS, ktorý je možné použiť na kódovanie reči ale aj audia. Skladá sa z dvoch kódérov. SILK, ktorý kóduje reč a CELT, ktorý je určený na kódovanie audia.

Rozdiel medzi rečovými a audio kodekmi je predovšetkým v tom, aký frekvenčný rozsah kódujú. U reči je to približne 300 Hz - 3,4 kHz pri vzorkovacej frekvencii 8 kHz, ktoré sa používa pri telefónoch, 50 Hz - 7 kHz pri vzorkovacej frekvencii 16 kHz pre širokopásmovú reč a u audia je to 20 Hz - 20 kHz do vzorkovacej frekvencie až 192 kHz[10]. U reči sa kladie dôraz predovšetkým na zrozumiteľnosť u audia na prirodzenosť zvuku. Od toho sa odvíjajú aj nároky na šírku pásma pri prenose sieťou.

Prvá kapitola obsahuje základný popis vlastností ľudského sluchu. Zahŕňa popis psychoakustického modelu, ktorý sa využíva pri stratovom kódovaní, prah počutia, ktorý hrá taktiež veľkú úlohu pri kompresii, kritických pásiem, ktoré sú ľudským sluchovým orgánom spracovávané a ďalším významným javom pri stratovej kompresii – maskovanie. V kapitole je taktiež stručne definovaná anomália, ktorá vzniká pri kódovaní v kodéri – pre-echo.

V druhej kapitole sú popísané základné princípy fungovania bezstratovej a stratovej kompresie. Z bezstratovej kompresie bol vybraný zástupca FLAC, ktorý je najrozšírenejší formát pri archivácii zvuku. U stratových formátov boli zvolené najrozšírenejší kodek – MPEG Layer 3 a OGG Vorbis. V kapitole sú zahrnuté základné vlastnosti.

Tretia kapitola pojednáva o rečových kodekoch, ich rozdelení na úzkopásmové a širokopásmové. Kapitola obsahuje taktiež popis lineárneho predikčného kódovania, súčasťou ktorého je lineárna predikcia, ktorá bola aj prakticky pomocou Matlabu overená. Záver kapitoly obsahuje popis kodéru CELP a najpoužívanejšieho rečového kodeku v IP telefónii, G.729. Taktiež sa tu nachádza popis kodeku OPUS s čiastkovými kodekmi CELT a SILK, ktoré v závislosti na nastavení využíva.

Kapitola štvrtá je zameraná na základný popis niektorých objektívnych metód porovnania zvukových kodekov. Ide o segmentové merania odstupú signál-šum a perceptuálne merania, predstaviteľmi ktorých sú Wide Spectral Slope meranie a Perceptual Evaluation of Speech Quality.

Posledná kapitola obsahuje porovnanie kodekov z hľadiska výpočtovej náročnosti a objektívnej miery. Zvolenými audio kodekmi boli MP3, OGG a FLAC. Pri nich bola zameraná špička využitia pamäte pri kódovaní pomocou kodérov lame a ffmpeg. Meranie prebiehalo v konzole systému OS Linux a bola meraná doba konverzie pri nastavení rôznej kvality. Rečové kodeky boli podrobené objektívnej metóde PESQ a boli zistené hodnoty MOS, ktoré vypovedajú o kvalite kodeku.

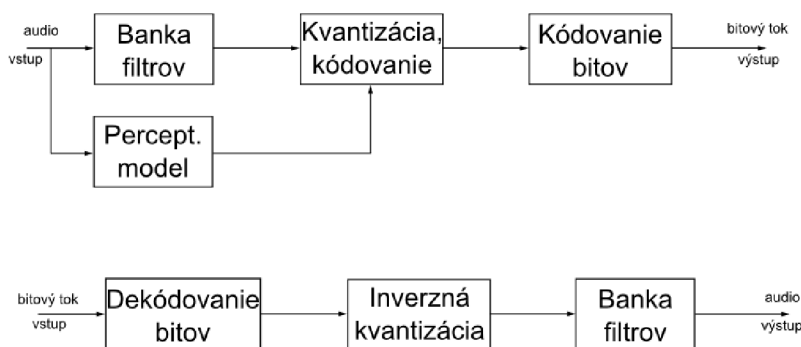


# 1 VLASTNOSTI ĽUDSKÉHO SLUCHU

V tejto kapitole sú popísané psychoakustický model a základné vlastnosti ľudského sluchu.

## 1.1 Psychoakustický model

Tento model sa používa pri stratovej kompresii, kde dochádza k vynechaniu alebo modifikácii niektorých detailov zvuku, na ktoré nie je ľudský sluch citlivý. Ide napríklad o orezanie frekvenčného pásma alebo maskovanie signálov. Využívajú sa *prah počutia* a *maskovanie*[6]. Kódovanie, ktoré využíva psychoakustický model, sa nazýva *perceptuálne kódovanie*. Jeho princíp je znázornený na obr. 1.1.



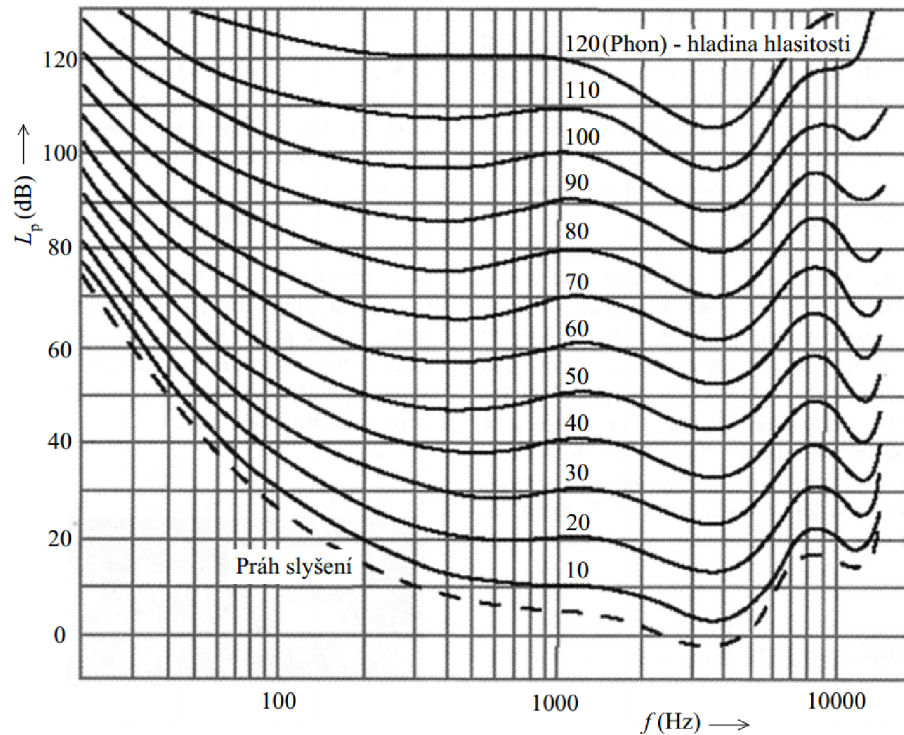
Obr. 1.1: Princíp perceptuálneho kodéru

Skladá sa z[2]:

- *banka filtrov*, kde sa signál prevedie do frekvenčnej alebo časovej oblasti,
- *perceptuálny model*, kde sa na základe psychoakustických pravidiel vypočíta signál, ktorý má byť na výstupe,
- *kvantizácia a kódovanie*, kde prebehne kvantizácia a kódovanie vrátane kvantizačného šumu,
- *kódovanie bitov*, výstupom ktorého je bitový tok, ktorý sa skladá z kvantizovaných a zakódovaných koeficientov a hlavičky (v závislosti na formáte).

Ľudský sluch je charakterizovaný prahom počutia, ktorý vyjadruje energiu potrebnú na vyvolanie tónu, ktorý bude v absolútne tichom prostredí poslucháčom počuteľný. Udáva sa v dB SPL. Jeho úroveň vzhľadom k frekvencii tónu nie je rovnaká. Tak isto sa krivka s vekom človeka mení. Znázorňuje ju obr. 1.2.

Sluchový orgán človeka má tvar slimáka. Zvukové vlny, ktoré do „slimáka“ dorazia sú spracovávané po pásmach – kritických pásmach. Je ich 24[25] a do 500 Hz je ich šírka 100 Hz, od 500 Hz sa každé pásmo rozšíri o 20% od predchádzajúceho[26].



Obr. 1.2: Krivky rovnakej hlasitosti a prah počutia[27]

Pri stratovej kompresii sa využíva maskovací jav. Vzniká pri výskyte dvoch alebo viacerých zvukov a spôsobuje, že niektorý z nich nie sme schopní vnímať. Maskovanie poznáme[20]:

- *Frekvenčné* – ľudský sluch nie je natoľko dokonalý, aby rozpoznať rozdiel napríklad medzi silným signálom s frekvenciou 1000 Hz a slabým signálom s frekvenciou 1005 Hz. Z tohto dôvodu je možné jeden tón vynechať.
- *Časové* – pri vnímaní hlasného tónu a jeho následnom ukončení trvá ľudskému uchu nejakú dobu (spravidla 10 ms), kým sa adaptuje na nové prostredie a je schopné vnímať aj tiché tóny.

Ďalej poznáme maskovanie *rovinné* a *priestorové*. V rovinnom sa berú do úvahy zdroje ležiace v rovine hlavy, u priestorového prichádza maskovaný zvuk z iného smeru ako maskovací[13].

Pri detekcii zvuku ľudským uchom vzniká ďalší zvukový artefakt – pre-echo, kedy je zvuk počuteľný skôr, ako zaznie.

## 2 AUDIO KODEKY

V tejto kapitole sú popísané základné princípy fungovania bezstratovej a stratovej kompresie, kodeky FLAC, MP3 a Vorbis, ktoré sú na niektorej z kompresí založené.

### 2.1 FLAC

Jedným z predstaviteľov bezstratových kodekov je Free Lossless Audio Codec – FLAC, vývoj ktorého začal Josh Coalson v roku 2000. Jeho prvá verzia bola vypustená 20. júla 2001[30]. Je voľne dostupný a podporovaný vo väčšine operačných systémov. Pojem bezstratový znamená, že výsledný zakódovaný zvuk nepostihne strata kvality, napríklad orezaním frekvenčného spektra alebo potlačením niektorých zložiek. Zo zakódovaných dát sa dajú získať dáta pôvodné. Táto možnosť sa deje ale na úkor nároku na úložný priestor, pretože bezstratová kompresia zmenší dáta oproti originálnym len o približne 5-20 %, kým stratová aj o 50-60 %[24].

FLAC je rýchly kodek, nevyžaduje veľké nároky na procesor. Podporuje pridávanie tagov skladbám, ilustráciu a rýchle prevíjanie. Je možné ho streamovať [31]. Veľkou výhodou je to, že nepodporuje ochranu proti kopírovaniu DRM – Digital Rights Management. Výhodou preto, že si môžeme zakúpenú hudbu kopírovať na viaceré zariadenia bez obmedzení.

#### 2.1.1 Bezstratová kompresia

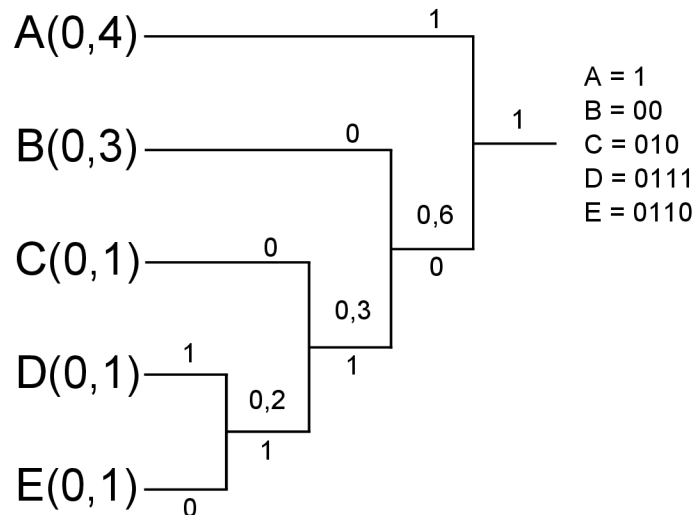
Pre bezstratovú kompresiu sa využívajú Lempel-Zivove a Huffmanovo kódovanie. Rozdiel spočíva v tom, že LZ kódovanie dokáže vytvoriť z dát premenlivej dĺžky dáta s dĺžkou fixnou. Oproti tomu, Huffmanovo kódovanie vytvára z fixne daného bloku dát dáta s variabilnou dĺžkou[14]. Mechanizmus bezstratovej kompresie sa neskladá zo zložitej blokovej schémy, pretože vstupné dáta sú len kódované a nie inak upravované.

#### Huffmanovo kódovanie

Huffmanovo kódovanie je založené na vedomosti početnosti výskytu jednotlivých znakov, ktoré je potrebné zakódovať. Znakom, ktorých pravdepodobnosť výskytu je menšia, sú zakódované vyšším počtom bitov ako znaky s vyššou pravdepodobnosťou výskytu[1].

Vytvorenie kódového slovníku môže prebiehať podľa obr. 2.1.

Pri každom znaku je udaná pravdepodobnosť výskytu. Zoradíme ich podľa veľkosti a najmenšie dve zlúčime dokopy. V tomto prípade sa jedná o znaky  $D$  a  $E$ .



Obr. 2.1: Príklad Huffmanovho kódovania

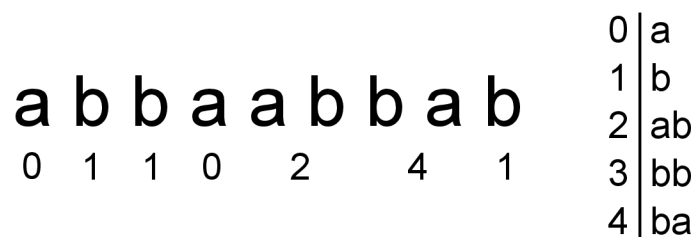
Vetve s nižšou pravdepodobnosťou priradíme 0 a s vyššou 1. Ďalej tieto znaky zlúčime so znakom *C* a pokračujeme až do konca, kým nevytvoríme strom. Kód znaku je potom určený cestou od koreňa k listu.

Úlohou dekodéru je z toku núl a jednotiek dekodovať jednotlivé znaky na základe kódovej knihy, ktorú musí dopredu poznať[18].

### LZ kódovanie

Lempel-Zivove kódovanie, na rozdiel od Huffmanovho, ktoré kóduje pevne dané bloky dát, je založené na postupnom vytváraní slovníka prúdu dát.

Princíp je zobrazený na obr. 2.2



Obr. 2.2: Príklad LZ kódovania

Máme danú postupnosť znakov *abbaabbab*. Vyberieme prvý znak *a*, zapíšeme ho do tabuľky a priradíme mu kód 0. Ďalším znakom je *b*. Kód pre tento znak v tabuľke nemáme, preto ho tam pridáme a dáme mu kód 1. Prvá dvojica znakov *ab* v tabuľke zaznamenaná nie je, preto ju tam tiež pridáme s kódom 2. Postupujeme ďalej pridelením kódu 1 tretiemu znaku v poradí *a* do tabuľky zaznamenáme *ba*

s kódom 3. Týmto spôsobom vytvárame slovník až do konca postupnosti, ktorú potrebujeme zakódovať.

V prípade, že príde do dekodéru postupnosť *0110241* vie dekodér podľa slovníku dekódovať pôvodnú postupnosť znakov.

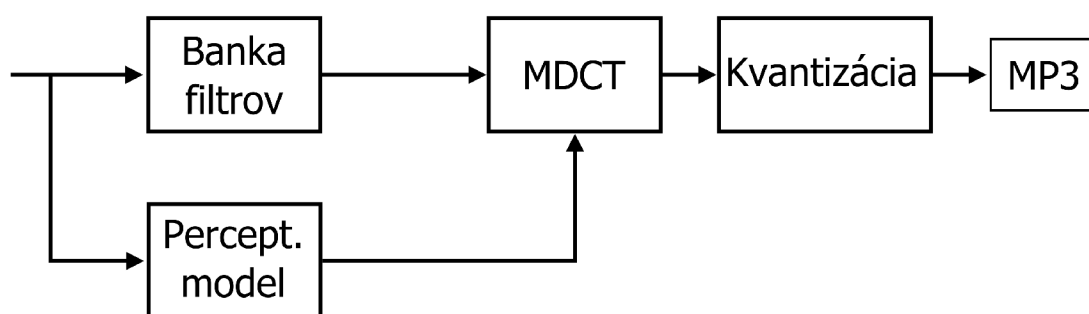
## 2.2 MPEG Layer III

Kodek MP3 je otvorený stratový kodek, ktorý bol vytvorený v roku 1991 prevažne na prehrávanie hudby v prenosných zariadeniach. V prípade, že chceme štandard implementovať, je potrebné zaplatiť za špecifikáciu. Hlavnou myšlienkou bolo zmenšiť veľkosť nahrávky a zjednodušiť dekodér natoľko, aby bol čo najmenej energeticky náročný.

Vzhľadom na jeho rozšírenie vznikli hardwarové a softwarové dekodéry. Sú súčasťou rôznych zariadení ako zvukové karty, prenosné prehrávače, televízie a pod.

### Kodér

Na obr.2.3 je zobrazený kodér MPEG Layer III. Vstupný signál je najprv rozdelený na samply, ktoré sú prepustené cez banku filtrov, kde sa signál rozdelí pásmovými priepustami na 32 subpásiem. Signál rovnako prejde perceptuálnym modelom, z ktorého je vedený výstup do bloku, kde prebehne MDCT. Každému subpásmu sa vypočíta SNR a za pomoci výstupu z psychoakustického modelu sa určí potrebný bitrate a vlastnosti maskovania. Nasleduje kvantizácia, Huffmanovo kódovanie a vytvorenie rámca v ktorom sú zakódované koeficienty a informácie potrebné na ich dekódovanie.



Obr. 2.3: Kodér MPEG Layer 3

Vzhľadom na širokú možnosť využitia MP3 bolo navrhnutých viac možností nastavení a každá implementácia tieto nastavenia využíva iným spôsobom. Niektoré z nich sú stavané na nižší bitrate, iné zas na vysoký (napr. LAME).

Kodéru je možné nastaviť jeden zo štyroch operačných módov:

- *Single channel* – inak nazývané aj mono. Nahrávka sa skladá z jedného kanálu ktorý je sám kódovaný.
- *Dual channel* – tento mód sa používa v nahrávkach, kde treba odlíšiť napríklad rôznu jazykovú stopu.
- *Stereo* – nahrávka sa skladá z dvoch na sebe nezávislých kanálov.
- *Joint stereo* – je to stereo, ktoré vzniká tak, že je nahratý len jeden mono kanál, z ktorého sa dopočítajú detaily pre ostatné kanály.

Vo všeobecnosti, v MPEG Layer III, sú dostupné nasledujúce vzorkovacie kmitočty: 8, 11,025, 12, 16, 22,05, 24, 32, 44,1 a 48 kHz.

V kodeku MP3 je možné nastaviť rôzne bitové hĺbky – 8 b (telefónna kvalita), 16 b (CD kvalita) a 24 b (DVD kvalita). Udáva počet napätových úrovní, ktorými je možné reprezentovať signál[21].

Mp3 používa dva typy bitrate – *konštantný* a *variabilný*. V konštantnom sa bitrate drží po celú dobu nahrávky na jednej úrovni, vo variabilnom sa mení v závislosti na náročnosti skladby. Tam, kde je zvuk zložitejší, zakóduje sa nahrávka vyšším dátovým tokom a v prípade ticha alebo nízkej hlasitosti sa použije nízky dátový tok.

Výhodou variabilného bitrate je úspora miesta, nevýhodou väčšia výpočetná náročnosť oproti konštantnému. V súčasnosti je najpoužívanejší dátový tok 128 kb/s, ktorý má zaručovať na poslech rovnakú kvalitu ako originál. Nie vždy je tomu tak, v závislosti napríklad od žánru hudby. Celá škála sa pohybuje od 8 do 320 kb/s[2].

## 2.3 Ogg Vorbis

Ogg Vorbis je slobodný formát, ktorý vytvorila nadácia Xiph.Org. Radíme ho medzi stratové a používa sa ako na kompresiu zvuku, tak reči[32]. Na rozdiel od mp3, ktorý podporuje maximálne 6 kanálov (5.1 zvuk), vorbis podporuje až 255 kanálov[23].

Ako väčšina stratových kodekov, tak isto Vorbis používa psychoakustický model. Kodér je jednoduchý a nenáročný na systémové prostriedky. Dekódovanie si nevyžaduje veľké množstvo procesorového času ale spotrebováva viac pamäte, pretože od začiatku procesu dekodovania používa kódový slovník, ktorý ďalej pribaľuje do hlavičky vorbisu pre ďalšiu fázu dekodovania. Vorbis sa stará len o samotné spracovanie signálu, nestará sa napríklad o framovanie, synchronizáciu ani ochranu proti chybám. Pre tento účel sa používa kontajner *ogg*. Kým u mp3 je možné nastaviť viac typov dátového toku, vorbis podporuje VBR (Variable BitRate).

## Algoritmus

Kódovanie zvuku do vorbisu prechádza určitým systémom[8].

1. MDCT a FFT – modifikovaná diskretná kosínová transformácia a rýchla Fourierova transformácia. Koefficienty sú transformované z časovej do frekvenčnej oblasti.
2. Psychoakustický model – vymažú sa nepočuteľné zložky.
3. Floor Mapping – lineárna predikcia.
4. Vymazanie Floor – frekvenčné spektrum sa rozdelí na časti podobné kritickým pásmam u sluchu človeka. Dosiahne sa vyhladenie spektra a redukcie kvantizačného šumu.
5. Spojenie kanálov – spojí pravý a ľavý kanál.
6. Vektorová kvantizácia.

## 3 REČOVÉ KODEKY

Pri kódovaní reči sa na vstupný signál použije D/A prevodník, ktorý vstupný reč nasegmentuje a kvantizuje. Kvantizácia je proces nevratný a určujeme u nej chybu – kvantizačný šum. Je závislý na použitom kvantizátore a na počte bitov, ktorými vzorku zakódujeme. Vysoký počet bitov má za následok veľké nároky na prenosovú kapacitu. Z toho dôvodu sa na redukciu prenosovej náročnosti používajú viaceré techniky.

Ludský hlas môže byť vo frekvenčnom pásme 0-8 kHz v závislosti na pohlaví jedinca a jeho veku. Kódovanie sa delí na *úzkopásmové* a *širokopásmové*. Úzkopásmové kódovanie prebieha vo frekvenčnom pásme 300 Hz až 3,4 kHz, ktoré zahŕňa najviac informácií reči. U širokopásmového je pásmo širšie, 50 Hz až 7 kHz. Na rozdiel od úzkopásmového obsahuje frekvencie nižšie ako 300 Hz, čo prispieva k lepšej prirodzenosti reči. Naopak frekvencie nad 3,4 kHz dodávajú reči zrozumiteľnosť. Pri širokopásmovom kódovaní reči sa kódované pásmo rozdelí na dve polovice – *nižšie* (0-6,4 kHz) a *vyššie* (6,4-7,2 kHz) subpásmo[15].

Kódovanie reči poznáme ďalej *škálovateľné*, a to vzhľadom na frekvenciu a dátový tok. U tohto typu kódovania je výstupný dátový tok rozdelený na vrstvy, z ktorých každá obsahuje nejaké informácie. Najdôležitejšia je základná vrstva, ktorá nesie najviac informácií. Ostatné vrstvy v spolupráci so základnou, ju dopĺňajú o detaily. Z tohto dôvodu je možné pri kódovaní použiť variabilný dátový tok (vypustíme vrstvy, ktoré nie sú pre daný segment dôležité).

**Škálovateľné kodéry dátového toku** zlepšujú reč v tom istom pásme, v ktorom je základná vrstva. Princípom je kódovanie rozdielu vstupnej reči a reči zo základnej vrstvy syntetizovanej.

**Škálovateľné kodéry šírky pásma** kódujú frekvenčné pásma, ktoré základná vrstva vynechala. Môžu fungovať aj nezávisle na základnej vrstve, ale s ňou sú efektívnejšie.

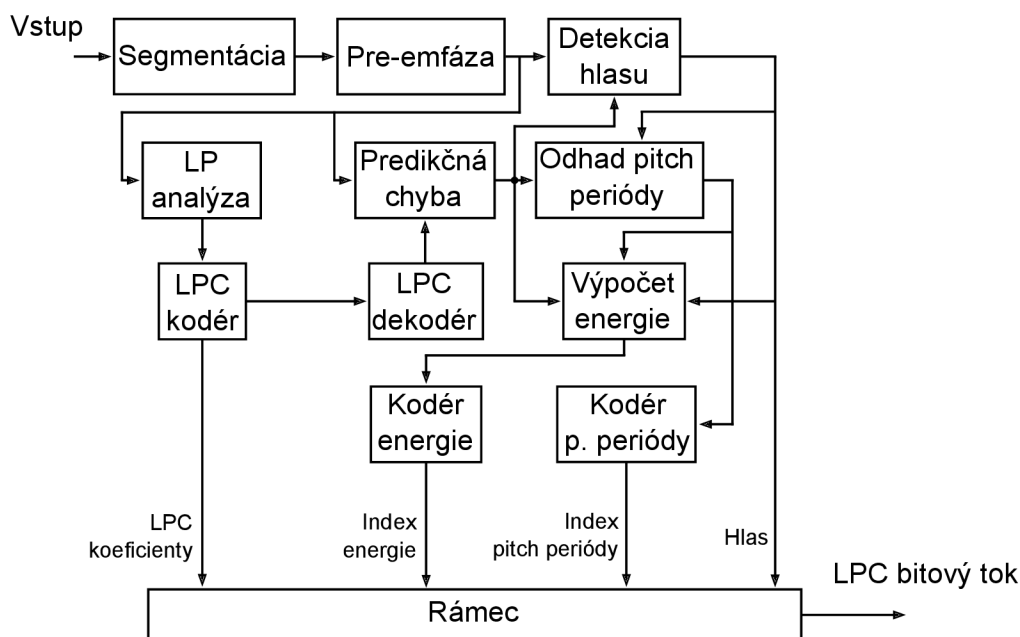
### 3.1 Linear predictive coding

Na obr. 3.1 je znázornené blokové schéma lineárneho predikčného kodéru.

Na vstup príde navzorkovaný signál, ktorý sa rozdelí na segmenty, s ktorými sa ďalej pracuje. Dĺžka segmentu môže byť napríklad 8 ms.

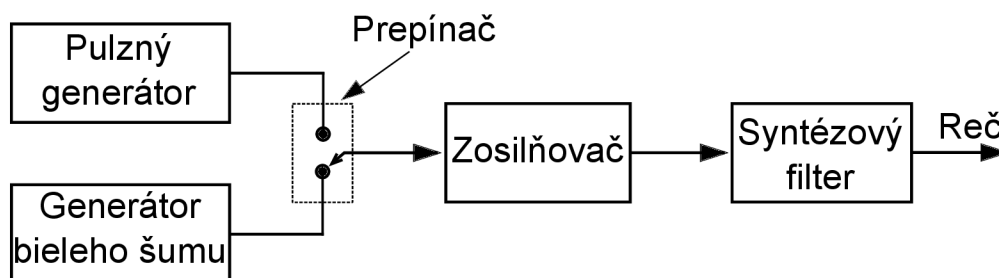
Segment vstúpi do preemfázového bloku, kde sú zosilnené jeho vyššie kmitočty. Výstup z preemfázového bloku smeruje do blokov detekcie hlasu, analýzy lineárnej predikcie a filtra predikčnej chyby[3].





Obr. 3.1: Bloková schéma LPC kodéru

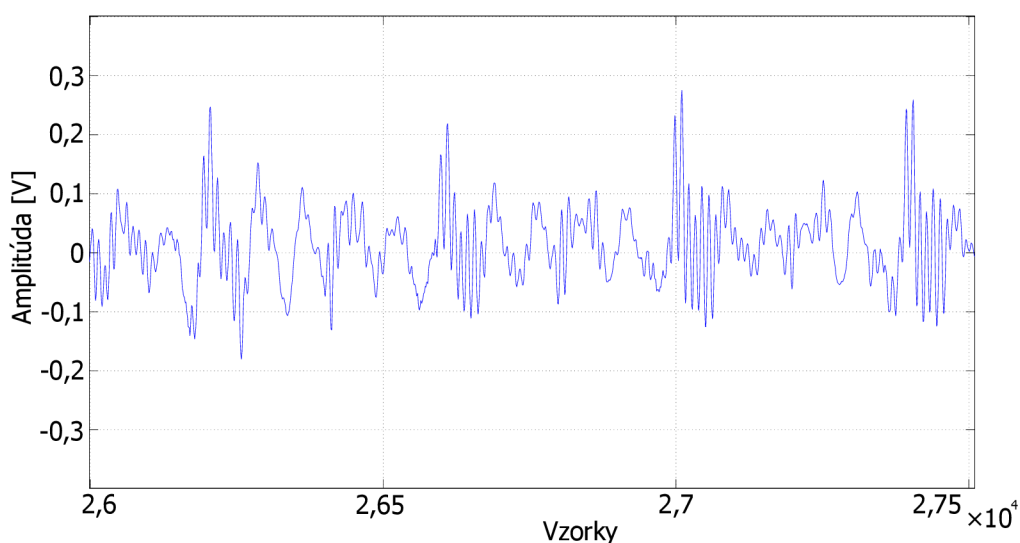
Detekcia hlasu spočíva v rozhodovaní, či sa na vstupe do tohto bloku nachádza signál hlasový alebo iný. Na obr. 3.2 je znázornené blokové schéma produkcie rečového signálu.



Obr. 3.2: Bloková schéma produkcie reči

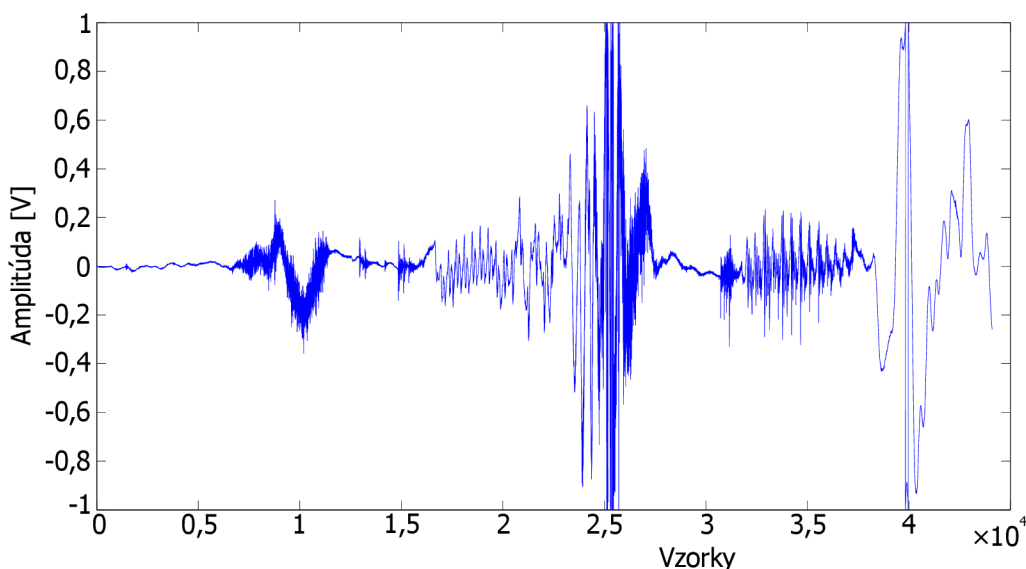
Vstup hlasového detektoru sa skladá z dvoch častí. Je to pulzný generátor a generátor bieleho šumu.

Ludský hlas vzniká prerušovaním prúdu vzduchu, idúceho z pľúc hlasivkami, ktorý je následne dotvorený jazykom, perami a mimikou. Prúd vzduchu predstavuje generátor bieleho šumu, vibrácie hlasiviek pulzný generátor. Hlasivky vibrujú určitou frekvenciou, ktorá sa mení s pohlavím a vekom človeka. Frekvencia hlasiviek slúži pre určenie tzv *pitch periódy*, ktorou je ďalej riadený pulzný generátor. Na obr. 3.3 je úsek rečovej nahrávky, ktorý reprezentuje práve hovorenú hlásku. Je možné vidieť, že isté krátke úseky sa opakujú a sú si podobné.



Obr. 3.3: Úsek zvukovej nahrávky reči

Na obr. 3.4 je zobrazené slovo „skúška“. Je vidieť, že charakteristika nehlasových prejavov, napr. „s“ a „k“, je iná, než hlasových. Je to tým, že človek pri vyrieknutí nehlasových zvukov nepoužíva hlasivky. Vzduch, idúci z pľúc, dotvárajú do výslednej formy len jazyk a zuby. Ďalej je možné si všimnúť, že v hláske „ú“ je viac energie než v „s“ a že jeho graf pretne nulu na osi  $x$  viackrát než u „ú“ [17]. V časoch 2,5 a 4 s je možné si všimnúť prebudenie signálu, ktoré vzniklo zlým nastavením citlivosti pri zachytávaní zvuku.



Obr. 3.4: Ukážka slova „skúška“

Ďalším blokom je prepínač, ktorý prepínaním vytvára signál podobný prúdu

vzduchu za hlasivkami, ktorý pokračuje do zosilňovača a syntézového filtra. Syntézový filter slúži ako obdoba ľudského jazyka, pier a mimiky. Na strane kodéru sa zisťujú parametre, ktoré sa posielajú ďalej a dekodér ich využije k syntéze reči. Tak isto sa vo filtri zachytí hustota spektra (PSD). Vďaka tomu je možné u dekodéru zložiť signál reprezentujúci hlas, pretože generátor bieleho šumu generuje náhodné čísla[11].

## 3.2 Lineárna predikcia

Základom moderného kódovania s nízkou prenosovou rýchlosťou a syntézy reči je lineárna predikcia. Jej východiskom je predpoklad, že aktuálny rámeček sa dá vypočítať pomocou lineárnej kombinácie predchádzajúcich rámečkov

$$s[n] = - \sum_{i=1}^M a_i \cdot s[n - i], \quad (3.1)$$

kde  $M$  je rád prediktora (obvykle sa volí medzi 10 a 20),  $s[n]$  je  $n$ -tá vzorka rečového signálu,  $u(n)$  je budiaca postupnosť a  $a_i$  sú lineárne predikčné koeficienty, ktoré špecifikujú daný rámeček. Lineárna predikcia taktiež odstraňuje položky, ktoré sa v signáli opakujú a tým sa zmenší veľkosť výsledných dát.

Počas výpočtu koeficientov vznikajú *predikčné chyby*, ktoré určujú odlišnosť pôvodného vzorku od odhadnutého

$$e[n] = s[n] - \hat{s}[n]. \quad (3.2)$$

Aktuálnym problémom je výpočet  $a_i$  koeficientov. Aby bola MSE (mean-squared error) čo najnižšia, musia sa vybrať správne predikčné koeficienty, napríklad priamou metódou[3]

$$\sum_{i=1}^M a_i \cdot R[i - k] = -R[k] \quad (3.3)$$

kde  $k = 1$  až  $M$ .

Výpočet môže byť riešený pomocou autokorelácie a prebieha tak, že sa zvolí vhodné veľké okno, v ktorom majú vzorky nenulovú hodnotu, mimo tohto okna sú vzorky rovné nule. Vzťah pre výpočet koeficientov je

$$a(i, k) = \sum_{n=0}^{N-1+(i-k)} y(n)y[n(i - k)], \quad (3.4)$$

kde  $(i - k)$  určuje lag[11].

Takto dostaneme maticu s rozmermi  $M \times M$ , ktorá je vyjadrená rovnicou

$$\mathbf{R}_s \cdot \mathbf{a}_i = -\mathbf{r}_s, \quad (3.5)$$

kde  $R_s$  môže byť Toeplitzova matica.

Výsledná sústava rovníc je zapísaná pomocou matice

$$\begin{bmatrix} R_s[0] & R_s[1] & \cdots & R_s[M-1] \\ R_s[1] & R[0] & \cdots & R_s[M-2] \\ \vdots & \vdots & \ddots & \vdots \\ R_s[M-1] & R_s[M-2] & \cdots & R_s[0] \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} = - \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_M \end{bmatrix} \quad (3.6)$$

Neznáme  $a_i$  zistíme pomocou vzťahu

$$\mathbf{a} = -\mathbf{R}_s^{-1} \cdot \mathbf{r}_s. \quad (3.7)$$

Implementáciu som volil v prostredí Matlab R2011b. Zdrojový kód programu je uvedený v prílohe A.

Na začiatku programu sa načíta ukážkový súbor `priklad.wav`, s ktorým sa ďalej pracuje. Jeho dĺžka je približne 1 sekunda a vzorkovacia frekvencia 8 kHz.

Signál je potrebné rozdeliť na segmenty (zvolil som 8 ms) s dĺžkou 64 vzoriek ( $8000 * 0,008 = 64$ ). Keďže signál nie je dlhý presne  $k * 64$  vzoriek, je potrebné posledný segment doplniť o nuly.

Následne sa vykoná transformácia vstupného vektoru vzoriek na maticu s počtom stĺpcov 64 a riadkov v závislosti na dĺžke nahrávky.

Nasleduje určenie pomocnej premennej pre výstupnú maticu koeficientov a vstupného lagu. Zvolil som ho 15, bežne je jeho veľkosť 10-20.

Cyklus s pevným počtom opakovaní má za úlohu vziať jeden segment s dĺžkou 64 vzoriek a podrobiť ho autokorelácii. O to sa stará funkcia `autokor`, ktorá vracia už hotovú Toeplitzovu maticu. Z navráteného vektoru `autokorelacia` je vypočítaný pomocný vektor s koeficientami  $a_i$  podľa vzťahu 3.7, ktoré sú ďalej uložené do výsledného vektoru `koef`.

Až vypočíta cyklus koeficienty pre všetky segmenty, vykoná sa ich transformácia do matice `LPkoef` s rozmermi  $M \times N$ , kde  $M$  je počet stĺpcov a je rovný počtu segmentov a  $N$  je počet riadkov rovný počtu koeficientov.

Funkcia `autokor` pracuje so vstupným segmentom *Segment* a lagom *lag*. Keďže indexovanie v Matlabe funguje od čísla 1 a nie 0, upravil som cyklus, ktorý počíta koeficienty od nuly do *lag-1*. Funkcia funguje na základe vzťahu 3.4, kedy sa do pomocného vektoru  $h$  uložia všetky prvky zo vstupného segmentu s príslušným skokom, ktorý určuje pomocná premenná  $i$ . Na začiatku je  $i$  rovné nule, preto sa na prvú pozíciu vektoru  $l$  uloží hodnota, ktorá sa vypočíta vynásobením prvkov segmentu s prvým prvkom tohto segmentu. Pre zvyšok výpočtu sa táto podmienka obchádza. Z výsledných hodnôt, uložených vo vektore  $l$  je následne vypočítaná Toeplitzova matica, ktorá slúži ako návratová hodnota tejto funkcie.

### 3.3 Code Excited Linear Prediction

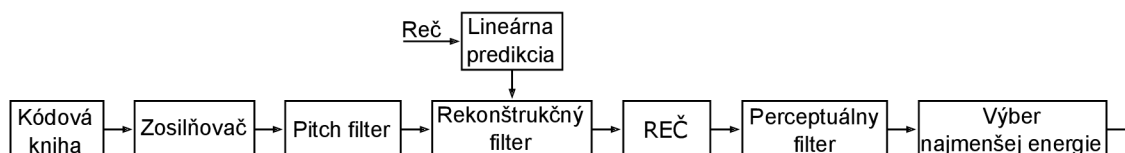
Kodér CELP je narozšírenejším kodérom reči. Je založený na LPC a využíva ho napríklad G.729. Úlohou kodéru je zistiť index zo segmentu signálu a poslať ho dekodéru, ktorý pomocou neho vyrobí signál.

Najpoužívanějšími variantami u internetovej aj bežnej telefónie sa stali Low Delay CELP (LD-CELP) a Algebraic CELP (ACELP). Rozlišujú sa v parametroch, ktoré zasielajú dekodéru a v pre-process a post-process filtroch.

Signál, ktorý CELP kodér spracováva, má vzorkovaciu frekvenciu 8 kHz a dátový tok 6,7 kbps. Kodér pracuje na systéme *analysis-by-synthesis*, čo znamená, že v kodéri sa signál najprv zakóduje, potom dekoduje a vyhľadajú parametre na minimalizáciu energie chybového signálu, ktoré sa pošlú dekodéru[9].

Voľba dĺžky segmentu závisí na implementácii. Tento segment je ďalej rozdelený na subsegmenty, v ktorých sa vyhľadávajú koeficienty kódovej knihy[22]. Kódová kniha je vytvorená napríklad náhodnými nenulovými hodnotami, ktoré sú oddelené nulami. Signál prichádzajúci z LPC bloku sa použije spolu s kódovou knihou pre syntézu reči, ktorá sa vo váhovacom filtri podľa istých pravidiel porovná s originálom a určí sa chyba. Váhovací filter je založený na perceptuálnom modeli[9]. Indexy koeficientov kódovej knihy, pre ktoré sú chyby subsegmentov najmenšie, sa ďalej spolu s veľkosťou zosilnenia a LPC koeficientami posielajú dekodéru. Výstupom kodéru je kódová kniha, vektory ukazateľov na koeficienty v tejto knihe a koeficienty s najnižšou energiou. Nevýhodami tohto kodéru sú vysoké oneskorenie, až 90 ms a nízka kvalita rekonštruovanej reči[22].

Na obr. 3.5 je znázornená bloková schéma CELP kodéru[9].



Obr. 3.5: Bloková schéma CELP kodéru

### 3.4 G.729

Kodér G.729 je založený na CELP s algebraickou štruktúrou s fixnou kódovou knihou. Pracuje s prenosovou rýchlosťou 8 kbps a poskytuje dobrú kvalitu zvuku. Medzi jeho základné vlastnosti patrí vzorkovacia frekvencia 8 kHz, 10ms dĺžka segmentu, oneskorenie 15 ms a Levinson-Durbinov algoritmus pre LPC filter[15].

Celý postup kódovania prebieha rozdelením segmentu na dva 5ms subsegmenty, pre ktoré sa vytvorí vlastná kódová kniha. V prvom segmente sú použité interpolované kvantizované a nekvantizované koeficienty lineárnej predikcie, v druhom segmente surové kvantizované a nekvantizované LP koeficienty. Pre subsegmenty sa vypočíta pitch perióda ktorá je ďalej poslaná dekodéru. Perióda je zakódovaná trinástimi bitmi (ôsmimi pre prvý subsegment, piatimi pre druhý subsegment), kódová kniha sedemnástimi a zosilnenie siedmimi bitmi na subsegment. LP koeficienty sú transformované na LSP (Line Spectrum Pairs) koeficienty a tie sú zakódované osemnástimi bitmi.

Dekódovanie prebieha na základe prijatých parametrov pre celý 10ms dlhý segment. Ide o LP koeficienty, dve čiastkové pitch periódny, dve kódové knihy a dva kódy pre zosilnenie. LSP koeficienty sú interpolované a pre každý subsegment prevedené na LP koeficienty. Reč je vytvorená kombináciou koeficientov kódovej knihy a koeficientov zosilnenia, ktoré sú ďalej filtrované lineárnym predikčným syntézovým filtrom. Výsledná syntezovaná reč je upravená post-processingovým filtrom – hornou priepustou[12].

## 3.5 OPUS

Stratový kodek OPUS bol navrhnutý skupinou IETF pre prenos rečového a audio obsahu pri konferenciách, ukladaní a streamovaní obsahu či bezdrôtové slúchadlá, reproduktory alebo mikrofóny. Zahŕňa funkcie SILK (Skype) a CELT (Xiph.Org) kodekov a je pod otvorenou licenciou[28].

Medzi jeho prednosti patrí vysoká prispôsobivosť na požiadavky,

- vzorkovacie kmitočty 8-48 kHz,
- konštantný aj variabilný bitrate,
- mono a stereo kanály,
- bitrate 6 (mono, reč)-510 (stereo, audio) kbps,
- algoritické oneskorenie 5-65,2 ms,
- dĺžku frame 2,5 až 60 ms.

Dokáže sa plynule (bez prerušenia) prispôbiť zmenám parametrov podľa nárokov zvuku alebo prenosového kanálu. V prípade, že do dekodéra, fungujúcom na  $f_{vz} = 48\text{ kHz}$  dorazí úzkopásmový signál, je schopný ho dekodovať.

OPUS dokáže kódovať rámce s dĺžkou 2,5, 5, 10, 20, 40 a 60 ms a dokáže ich aj spájať viacero do jedného paketu v prípade, že zdieľajú spoločné parametre ako bitrate, počet kanálov atď. Paket môže mať dĺžku maximálne 120 ms (max 1275 B) a začína záhlavím (TOC – table of contents) s veľkosťou 1 B, ktoré signalizuje nastavenia paketu. Na výber je možných 32 operačných režimov, šírka pásma a veľkosť

rámca. Východným výberom dĺžky rámca je 20 ms.

Problémom pri prenose dát prenosovým kanálom je chybovosť. Tá je u tohto kodeku riešená mechanizmom FEC (Forward Error Correction), ktorý funguje tak, že významné znaky reči, ako napríklad prízvuk, zakóduje znovu s nižším bitrate.

Kodek OPUS dokáže efektívne využívať prenosové pásmo aj tým, že znižuje bitrate v prípade šumu na pozadí alebo ticha. Takto je každých 400 ms zakódovaný jediný rámec. Tento mechanizmus sa nazýva DTX (Discontinuous Transmission).

Môže fungovať v troch režimoch:

- samotný SILK – pre reč,
- samotný CELT – pre audio,
- hybridný – kombinácia SILK a CELT pre širokopásmovú reč.

### Hybridný

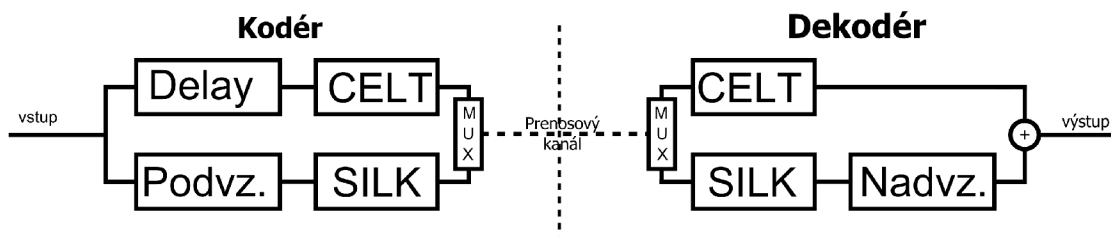
Dĺžka rámca je 10 alebo 20 ms pre super-wide bandwidth alebo full bandwidth. Každý frame je rozdelený na signály s nízkym a vysokým kmitočtom na hranici 8 kHz. Signály s nižším kmitočtom sú kódované lineárnou predikciou, signály s vyššou frekvenciou sú kódované pomocou MDCT (signály pod 8 kHz sú zahadzované).

Vzorkovací kmitočet je možné vybrať samostatne na strane kodéru aj dekodéru. To umožňuje plnú kompatibilitu nezávisle na hardware. LP vrstva pracuje s dvojnásobným vzorkovacím kmitočtom oproti audio, maximálne však 16 kHz, ktoré sa používa už v audio s väčšou šírkou pásma. Vrstva MDCT funguje vždy so vzorkovacím kmitočtom 48 kHz.

Výsledok SILK a CELT dekodérov je spojený do rámca ktorý sa prenesie dekodéru. Vstup CELT kodéru je oneskorený o 2,7 ms, čím je zabezpečené synchronné dorazenie oboch zložiek. Toto oneskorenie je možné znížiť, avšak na úkor kvality.

Obidve vrstvy používajú ten istý entropický kódér. Napriek tomu, že LP vrstva produkuje VBR a MDCT vrstva CBR, chýbajúce bity sú doplnené a výsledkom je konštantný bitrate.

Na obr.3.6 sú zobrazené kodér a dekodér, v ktorých je možné vidieť použitie SILK aj CELT blokov.



Obr. 3.6: Kodér a dekodér

### 3.5.1 SILK

SILK je rečový kodek navrhnutý spoločnosťou Skype založený na lineárnej predikcii. Dokáže meniť parametre kódovania priamo počas streamu zvukových dát. Bol navrhnutý pre variabilný bitrate, avšak kodér dokáže produkovať len konštantný bitrate. Verzia tohto kodeku použitá v OPUSe nie je referenčná, je vždy trochu pozmenená oproti východzej verzii[29].

Môže fungovať v štyroch režimoch vzorkovacieho kmitočtu:

- *úzkopásmový* – 8 kHz,
- *strednopásmový* – 8 alebo 12 kHz,
- *širokopásmový* – 8, 12 alebo 16 kHz,
- *super-širokopásmový* – 8, 12, 16 alebo 24 kHz

a v prípade potreby ho v reálnom čase prispôsobí.

Ďalšou vlastnosťou je prispôsobenie časového rámca. SILK umožňuje do jedného TCP/UDP paketu skombinovať viac časových rámcov a tým znížiť nároky na šírku pásma. Nevýhodou je ale vyššia náchylnosť na chyby pri prenose a latencia.

Bitrate tohto kodeku sa môže pohybovať od 6 do 40 kbps. Zvýšením bitrate sa znižuje kvantizačný šum. SILK obsahuje mechanizmus DTX (Discontinuous Transmission), ktorý zisťuje dlhodobé ticho alebo šum a ten kóduje jedným bitom v časovom rámci 400 ms.

Ako ochrana proti strate paketov bol vytvorený mechanizmus FEC (Forward Error Correction). Jeho princíp spočíva v opätovnom kódovaní dôležitých prvkov reči, napríklad prízvuku, na nižší bitrate a táto informácia je pridaná nasledujúcemu paketu. Signálu, ktorý vstúpi do dekodéru sú rozlíšené parametre, ktoré sú ďalej použité na vytvorenie excitovaného signálu. Signál sa ďalej v prípade požiadavky parametru rozdelí na stereo a nadvzorkuje na 48 kHz a v prípade použitia hybridného módu sčíta s výstupom z CELT dekodéru.

### 3.5.2 CELT

Constrained Energy Lapped Transform je kompresný kodek s otvoreným zdrojovým kódom (BSD) určený predovšetkým pre audio. Medzi jeho vlastnosti patrí veľmi nízka latencia (nastaviteľná 1,3-24 ms), vzorkovací kmitočet 8-48 kHz, podpora stereo, konštantný bitrate 32-128 kbps, ale aj vyššie. Oneskorenie tohto kodeku je 2,5 ms kvôli prekryvaniu okien u MDCT[28].

V prípade širokopásmového alebo menej kvalitného audia sa nepoužíva MDCT vrstva. U klasického audia sa MDCT vrstva používa.



## 4 HODNOTENIE KVALITY

Porovnávaním vecí, zariadení vzniká istá miera konkurencie. Rovnako to funguje aj u rečových i audio kodekov. Každý kodek sa snaží byť pre svoj účel čo najefektívnejší. Veľký dôraz sa kladie na prenosové nároky a kvalitu. Práve pre porovnanie kvality existujú viaceré druhy subjektívnych a objektívnych mier.

Subjektívne metódy sa opierajú o ľudský sluch. Pre návrh kodeku sú dôležitejšie práve subjektívne miery. Ich nevýhodou je časová a finančná náročnosť. Je potrebné vymyslieť audiotest ktorý je potom predvedený určitej skupine poslucháčov. Tieto merania ale nevykazujú prílišnú hodnotu, pretože ľudský sluch je ovplyvnený denným časom, fyzickou vyťaženosťou poslucháča alebo prostredím, v ktorom sa nachádza.

Oproti tomu objektívne miery sú menej náročné na prostriedky a ich výsledky sú nemenné. V tejto kapitole im bude venovaná pozornosť. Ide o segmentové meranie SNR, meranie odlišností spektra založené na lineárnej predikcii a perceptuálne orientované merania.

Pri objektívnom meraní reči sa vstupný signál najprv navzorkuje na rámce s dĺžkou 10-30 ms a potom sa počíta skreslenie medzi aktuálnym a originálnym signálom. Celková miera skreslenia sa vypočíta pre všetky rámce signálu a potom sa priemeruje. Meranie skreslenia je možné urobiť v časovej (napr. odstup signálu od šumu – SNR) aj frekvenčnej oblasti (napr. meranie vzdialenosti spektra LPC).

### 4.1 Segmentové merania SNR

Segmentové meranie odstupu signálu od šumu je možné merať v časovej aj frekvenčnej oblasti.

V časovej oblasti prebieha výpočet porovnaním čistého signálu  $x[n]$  a upraveného  $\hat{x}[n]$  v segmentoch s dĺžkou  $N$  (zvyčajne sa volí 15-20 ms) a počtom segmentov  $M$  podľa vzťahu

$$SNR_{seg} = \frac{10}{M} \sum_{m=0}^{M-1} \log_{10} \frac{\sum_{n=N \cdot m}^{N \cdot m + N - 1} x^2[n]}{\sum_{n=N \cdot m}^{N \cdot m + N - 1} (x[n] - \hat{x}[n])^2}. \quad (4.1)$$

Preďtým je ale potrebné signály zarovnať v čase. Veľkým problémom sú oblasti ticha v nahrávke, ktoré majú veľmi malú energiu a preto negatívne ovplyvňujú výsledok merania. Preto je potrebné tieto časti odstrániť alebo upraviť, napríklad vyhladením  $SNR_{seg}$  hodnôt.

Vzhľadom k tomu, že v hore popísanom prípade, sa energia ticha v nahrávke blíži

k zápornému nekonečnu, pán Richards[16] upravil rovnicu 4.1 do tvaru

$$SNR_{\text{segr}} = \frac{10}{M} \sum_{m=0}^{M-1} \log_{10} \left( 1 + \frac{\sum_{n=N \cdot m}^{N \cdot m + N - 1} x^2[n]}{\sum_{n=N \cdot m}^{N \cdot m + N - 1} (x[n] - \hat{x}[n])^2} \right), \quad (4.2)$$

kedy je možné do výsledku započítať aj segmenty ticha. V tomto prípade je výsledok záporného nekonečna eliminovaný a hodnota SNR segmentu môže dosiahnuť najnižšiu hodnotu nula.

Segmentový SNR je možné rozšíriť do frekvenčnej oblasti podľa vzťahu

$$fwSNR_{\text{seg}} = \frac{10}{M} \sum_{m=0}^{M-1} \log_{10} \frac{\sum_{j=1}^K B_j \log_{10} \frac{F^2(m,j)}{F(m,j) - \hat{F}(m,j)}}{\sum_{j=1}^K B_j}, \quad (4.3)$$

kde  $B_j$  je váha  $j$ -tého frekvenčného pásma,  $K$  je počet pásiem,  $M$  je celkový počet segmentov v signále,  $F(m, j)$  je amplitúda banky filtrov originálneho signálu v  $j$ -tom frekvenčnom pásme na  $m$ -tom segmente a  $\hat{F}(m, j)$  je amplitúda banky filtrov upraveného signálu v rovnakom pásme. Výhodou merania vo frekvenčnej oblasti je možnosť priradenia váh rôznym kritickým pásmam spektra[4].

## 4.2 Perceptuálne orientované merania

Perceptuálne merania sú vierohodnejšie, pretože sú založené na odhade ľudskej reči. V tejto podkapitole budú niektoré bližšie popísané.

### 4.2.1 WSS meranie

Weighted Spectral Slope meranie spočíva v meraní váhových rozdielov spektra dvoch segmentov v kritických pásmach.

Najprv je potrebné pre každé pásmo vypočítať energiu

$$\Delta E_s(f) = E_s(f + 1) - E_s(f), \quad (4.4)$$

$$\Delta E_{\hat{s}}(f) = E_{\hat{s}}(f + 1) - E_{\hat{s}}(f), \quad (4.5)$$

kde  $E_s(f)$  je energia originálneho signálu,  $E_{\hat{s}}(f)$  energia spracovaného signálu. Následne je určená najbližšia špička  $P(f)$  pre  $\Delta E(f)$  väčšie a menšie od nuly. Potom sa spočíta váha v každom pásme

$$W(f) = \frac{W_s(f) + W_{\hat{s}}(f)}{2}, \quad (4.6)$$

kde

$$W_s(f) = \frac{20}{20 + E_{s,max} - E_s(f)} \frac{1}{1 + P_s(f) - E_s(f)} \quad (4.7)$$

$$W_{\hat{s}}(f) = \frac{20}{20 + E_{\hat{s},max} - E_{\hat{s}}(f)} \frac{1}{1 + P_{\hat{s}}(f) - E_{\hat{s}}(f)}. \quad (4.8)$$

Nakoniec výsledná hodnota merania je

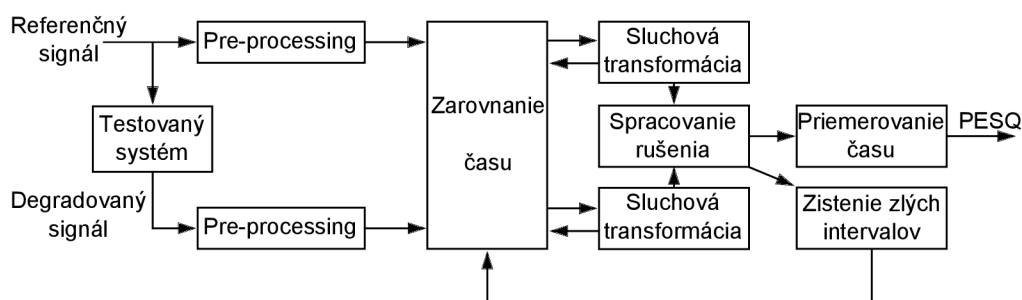
$$WSSM = \frac{1}{N} \sum_{k=1}^N \frac{\sum_{f=1}^{24} W(f) [\Delta E_s(f) - \Delta E_{\hat{s}}(f)]^2}{\sum_{f=1}^{24} W(f)}. \quad (4.9)$$

Čím je hodnota WSSM menšia, tým vyššia je kvalita reči[5].

## 4.2.2 PESQ

Perceptual Evaluation of Speech Quality je meranie, ktoré na rozdiel od predchádzajúcich do seba zahŕňa aj skreslenia, ktoré vznikajú pri prenose sieťou – strata paketov, oneskorenie signálu a skreslenie kodéru. Využíva perceptuálne meranie kvality reči *PSQM*.

Na obr. 4.1 je zobrazené blokové schéma výpočtu PESQ.



Obr. 4.1: Princíp PESQ

Signál, ktorý dorazí z vedenia, musí byť zosilený na štandardnú počutelnú úroveň. Deje sa tak v *pre-processing* bloku, ktorý je zložený zo zosilňovačov, ktoré sú vypočítané na základe *rms* hodnôt reči prechádzajúcej pásmovou priepustou (350-3250 Hz). Následne je signál filtrovaný filtrom s odozvou podobnou telefónnemu prístroju.

Vyššie upravený signál dorazí do bloku *zarovnanie času*, kde sa spočíta časový rozdiel medzi originálnym a degradovaným signálom. Najprv sa spočíta krížová korelácia originálneho aj skresleného signálu a vďaka nej sa aktívne segmenty rozdelia na podsegmenty. Výsledkom operácií sú doby oneskorenia, ktoré sa v priebehu signálu menia.

V *sluchovej transformácii* je signál mapovaný a je upravená jeho úroveň a frekvenčné spektrum. Podrobnejšie informácie je možné nájsť v [4].

## 5 POROVNANIE ZVUKOVÝCH KODEKOV

Pre porovnanie kodekov som zvolil prevod medzi formátmi a to konkrétne z WAV do príslušných z tab. 5.1. Zdrojovými súbormi boli skladby rôznych žánrov a dĺžiek. Prvá skladba je kludná muzika od U2 (4:58), druhá rýchla gitarová od Sum 41 (3:14) a posledná rýchla štýlu Drum and Bass od Qemists (5:37). Cieľom tohto testu bolo zistiť časovú závislosť kódovania od nastavenej výslednej kvality.

K porovnaniu som vyžíval počítač s programom *Free Audio Converter* pod operačným systémom Windows 7, ktorému som v správcovi úloh pridelil jedno jadro procesoru a nastavil najväčšiu prioritu. V programe som si vytvoril vlastné profily, ale používal som aj niektoré prednastavené. Dobu konverzie som meral ručnými stopkami. Každý výsledný čas je výsledkom priemeru minimálne piatich hodnôt získaním niekoľkonásobným meraním.

Táto metóda ale nie je príliš presná, pretože tu vzniká množstvo chýb merania, ale pre orientáciu postačí. Jednou z nich je fakt, že reakčná doba človeka je niekoľko desiatok milisekúnd a je rozdielna pri sluchovom a zrakovom vnímaní. Preto som pri meraní využil radšej sluch, pretože sa reakčná doba podstatne zníži. Vykonal som to tak, že som si v programe nastavil zvukové upozornenie po dokončení konverzie a po jeho zaznení som stopky zastavil. Ďalšia chyba vzniká z dôvodu uchovávanía dát v cache pamäti procesora. Predpokladám, že program nemá v sebe zabudovanú funkciu na jej vymazanie po každej operácii.

U všetkých nastavení som bral do úvahy stereo a vzorkovaciu frekvenciu 44,1 kHz, keďže pôvodné skladby mali dva kanály a daný kmitočet vzorkovania. U formátu MP3 som si zvolil nasledujúce nastavenia bitrate:

- *32 kbps*, najnižší nastaviteľný,
- *128 kbps*, ktorý má podľa špecifikácie zaručovať kvalitu nerozlišiteľnú od originálu,
- *192 kbps*, ktorý zo subjektívneho hľadiska považujem za najlepšiu voľbu pri konverzii,
- *320 kbps*, najvyšší možný nastaviteľný. U tohto dátového toku som zvolil tiež nastavenie *joint stereo*, pretože ma zaujímal časový rozdiel konverzie oproti klasickému stereu,
- *VBR V2*, ktorý predstavuje premenlivý dátový tok s dobrou kvalitou (170-210 kbps),
- *VBR V0*, ktorý predstavuje premenlivý dátový tok s najlepšou kvalitou (220-260 kbps).

Pre OGG som zvolil:

- *48 kbps*, najnižší nastaviteľný,

- 112 kbps, ktorý ma podľa špecifikácie zabezpečiť kvalitu zrovnateľnú so 128kbps MP3,
- 128 kbps, bitrate rovnaký ako u MP3,
- 192 kbps, z rovnakého dôvodu,
- 320 kbps, najvyšší možný nastaviteľný.

U formátu FLAC som zvolil jedno nastavenie a to 857 kbps a 48 kHz vzorkovaciu frekvenciu.

Ako prvú vec som si všimol rozdiel doby konverzie skladieb do formátu MP3 s dátovým tokom 320 kbps medzi stereom a joint stereom. Je vidieť, že je výpočtová náročnosť u joint sterea vyššia vo všetkých prípadoch. Rovnako aj úspora miesta, ktoré zaberie výstupný súbor s premenlivým dátovým tokom, je na úkor času.

Ďalej ma zaujímal rozdiel doby konverzie 128kbps MP3 a 112kbps OGG, ktoré majú zaručovať rovnakú kvalitu. Čas, potrebný na konverziu zdrojových skladieb do OGG bol v dvoch prípadoch o niečo vyšší ako u MP3.

Zaujímavosťou je, že mala doba konverzie do formátu OGG menšiu odchýlku ako do MP3. Kým konverzia prvej skladby do najnižšej kvality MP3 trvala 8,52 s, do OGG to bolo až 13,98 s.

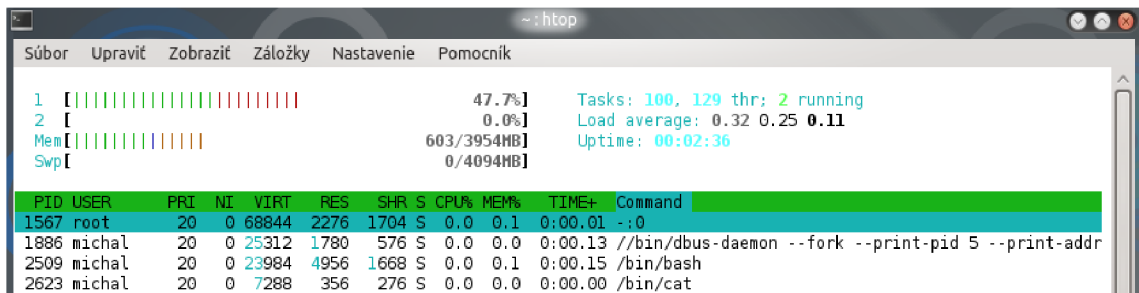
Tab. 5.1: Časy [s] potrebné na prevod skladieb do formátov

Kodek a nastavenie (bitrate [kbps], vzorkovací kmitočet [kHz])		Číslo skladby		
		1	2	3
<b>MP3</b>	32, 44,1, stereo	8,52	7,78	10,83
	128, 44,1, stereo	14,07	10,67	15,21
	192, 44,1, stereo	10,76	7,04	14,28
	320, 44,1, stereo	12,04	7,17	14,28
	320, 44,1, joint. s	14,49	8,19	14,45
	VBR V2	18,28	9,81	18,33
	VBR V0	15,67	9,66	16,99
<b>OGG</b>	48, 44,1, stereo	13,98	9,07	16,56
	112, 44,1, stereo	14,94	9,63	17,77
	128, 44,1, stereo	14,77	9,54	17,09
	192, 44,1, stereo	14,37	9,45	17,62
	320, 44,1, stereo	15,61	11,76	2,29
<b>FLAC</b>	857, 48, stereo	3,48	2,49	4,56

## 5.1 Pamäťová a časová náročnosť audio kodekov

Prípravou na tieto testy bolo vyradenie jedného jadra procesoru z multitaskingu (príznak *isolcpus=1* pri štarte systému) a pridelenie *afinity* procesu dekódovania tomuto jadru. Využíval som operačný systém Ubuntu 12.04.

Na obr.5.1 je možné vidieť správnu funkčnosť izolovania jadra. Logický procesor číslo dva nie je vôbec systémom využívaný. Tiež bolo potrebné vypnúť úsporný režim procesoru v bios, pretože doba zmeny frekvencie by sa mohla prejaviť na výsledkoch.



Obr. 5.1: Systém v stave idle

Nasledujúcim krokom bolo spustenie procesu kódovania a pridelenie afinity druhému jadru. Kódovanie som spustil príkazom

```
time taskset 02 lame spor.wav -b 128k spor.mp3
```

kde *time* meria čas, ktorý zabral proces procesoru, *taskset 02* pridelí afinitu druhému logickému procesoru a zvyšok značí použitie kodéru lame s výstupným bitrate 128 kbps na vstupnú skladbu *spor.wav*. Ilustrácia procesu kódovania je zobrazená na obr.5.2.

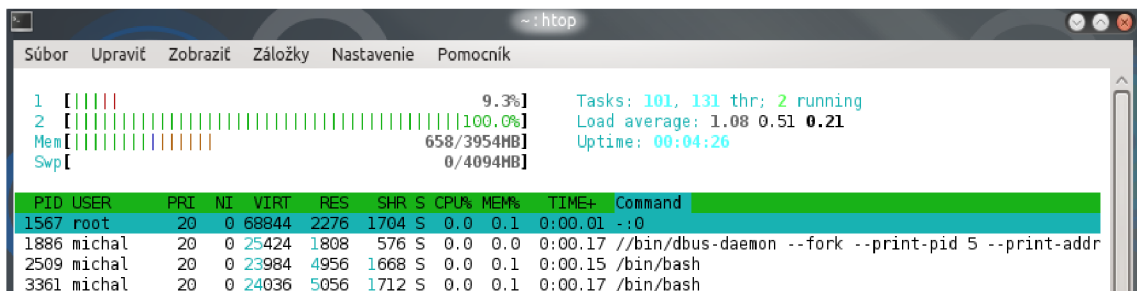
```
michal@c03-630b:~/diplomka$ time taskset 02 lame spor.wav -b 128k spor.mp3
LAME 3.99.3 64bits (http://lame.sf.net)
Using polyphase lowpass filter, transition band: 16538 Hz - 17071 Hz
Encoding spor.wav to spor.mp3
Encoding as 44.1 kHz j-stereo MPEG-1 Layer III (11x) 128 kbps qual=3
  Frame      | CPU time/estim | REAL time/estim | play/CPU |   ETA
4900/14458 (34%)| 0:04/ 0:12| 0:04/ 0:12| 29.091x| 0:08
-----04:09-----
kbps      LR  MS %      long switch short %
128.0     13.2 86.8      88.5 5.9 5.6
```

Obr. 5.2: Proces kódovania wav do mp3

V prílohe B sa nachádza grafické znázornenie využitia pamäte pri kódovaní z wav do mp3 a ogg. Využíval som príkazy:

- `valgrind --tool=massif --time-unit=B --massif-out-file=massif.ogg .32-64 oggenc -m 32 -M 64 spor.wav`
- `valgrind --tool=massif --time-unit=B --massif-out-file=massif ffmpeg.32 ffmpeg -i spor.wav -y -acodec libmp3lame -ab 32k sporr.mp3`
- `valgrind --time-unit=B --tool=massif lame spor.wav -b 320k spor.mp3`

Pridelenie spracovania procesu druhému jadru je možné overiť pomocou obr.5.3, kde zataženie druhého logického procesoru je na 100 percentách.



Obr. 5.3: Systém v záťaži

Pri kódovaní pomocou kodéru flac som použil príkaz

```
time taskset 02 flac -1 spor.wav
```

kde *time* označuje meranie času, *taskset 02* pridelenie procesu druhému (vyhradenému) jadru procesora, *flac -1* flac kodér s použitím najnižšej miery kompresie a *spor.wav* je zdrojová audio nahrávka.

Tab. 5.2: Priemerné časy pre flac

FLAC			
Kvalita	1	5	8
	0,9432 s	2,164 s	7,2432 s

Pre kódovanie do formátu MP3 som využíval kodéry *ffmpeg* a *lame*. U *ffmpeg* kodéru som použil

```
time taskset 02 ffmpeg -i spor.wav -y -acodec libmp3lame -ab 32k sporr.mp3
```

kde využívam knižnicu od *lame*, vstupný súbor zostáva rovnaký, bitrate audia je 32kbps a výstupný súbor je *sporr.mp3*. Časová záťaž je zobrazená v tab. 5.3.

Tab. 5.3: Priemerné časy pre ffmpeg

<b>ffmpeg</b>			
Kvalita	32 kbps	128 kbps	320 kbps
	8,2898 s	9,793 ms	9,6858 s

Pri kodéri lame som použil

```
time taskset 02 lame spor.wav -b 320k spor.mp3
```

kde je nastavený bitrate na 320 kbps. V tab. 5.4 sú uvedené priemerné časy potrebné pre zakódovanie nahrávky do rôznej kvality.

Tab. 5.4: Priemerné časy pre lame

<b>lame</b>			
Kvalita	32 kbps	128 kbps	320 kbps
	6,7592 s	12,905 s	12,8034 s

V tab. 5.5 je časová náročnosť kodéru vorbis pre rôzne kvality. Použitý príkaz bol

```
time taskset 01 oggenc -b 48 spor.wav
```

pre východzí variabilný bitrate, ktorý si kodér určil sám, alebo

```
time taskset 01 oggenc -m 32 -M 64 spor.wav
```

kde *-m 32 -M 64* značí použitie minimálneho bitrate 32 a maximálneho 64 kbps.

Tab. 5.5: Priemerné časy pre vorbis

<b>vorbis</b>				
Kvalita	48 kbps	112 kbps	224 kbps	500 kbps
	10,0402 s	10,5962 s	10,4898 s	11,757 s

Posledným testovaným kodekom bol OPUS. Spustenie úlohy som realizoval

```
time taskset 02 ./opus_demo -e audio 48000 2 32000 spor.wav spor
```

kde *-e* značí použitie iba kodéru, 48000 je vzorkovací kmitočet, 2 kanály (stereo) a 32 kbps bitrate. Výsledky časovej náročnosti sú v tab 5.6.



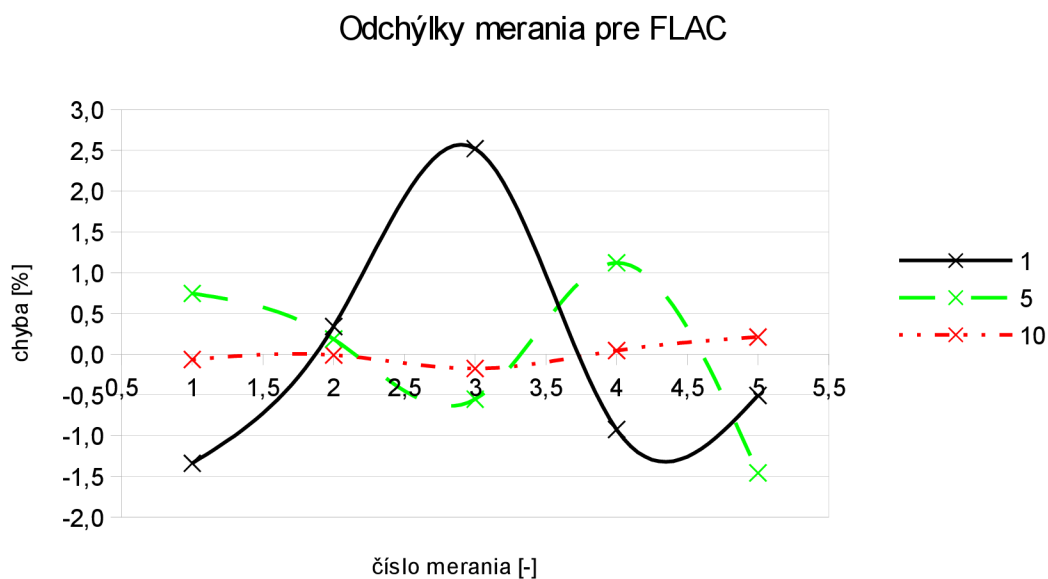
Tab. 5.6: Priemerné časy pre OPUS

vorbis			
Kvalita	32 kbps	128 kbps	320 kbps
	4,74 s	6,598 s	8,211 s

### 5.1.1 Odchýlky merania

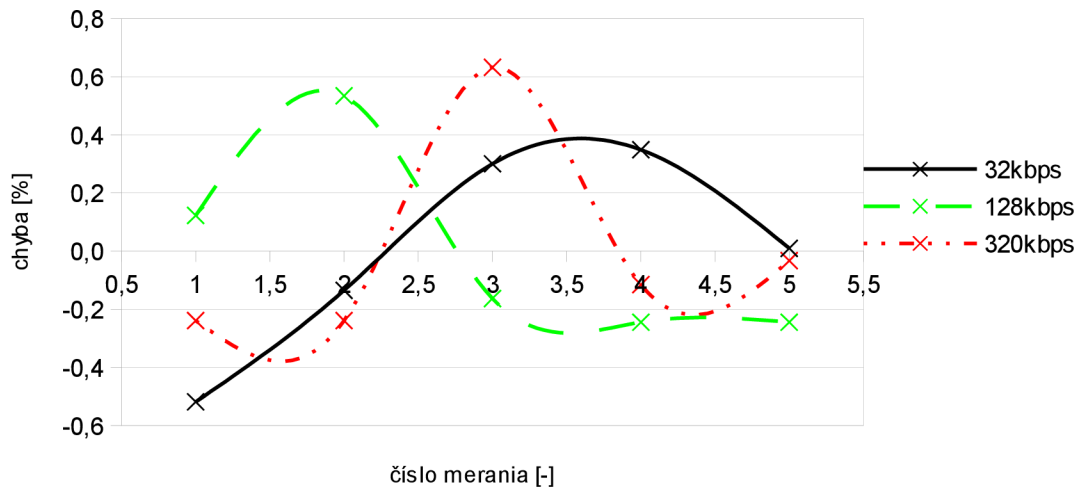
V nasledujúcich grafoch sú znázornené odchýlky merania. Meraný čas bol užívateľský, ktorý podľa manuálu programu *time* nezahŕňa do merania komplet celý proces aj so systémovými volaniami ale len funkcie spusteného programu.

Meranie prebehlo pri každom nastavení kvality päťkrát, následne bol vypočítaný priemerný čas a z neho odchýlky. Z grafov je možné si všimnúť, že odchýlky sú minimálne, okrem prípadu kódovania do formátu flac pri najnižšej kompresii, kedy kódovanie trvalo necelú sekundu a preto je odchýlka väčšia než 1 %.



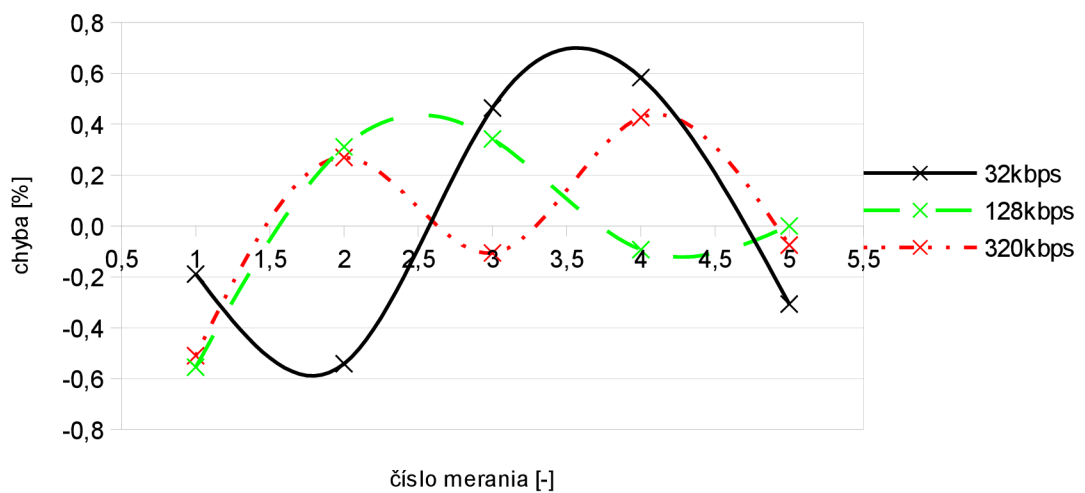
Obr. 5.4: Odchýlky merania kodeku flac

### Odchýlky merania pre FMPEG



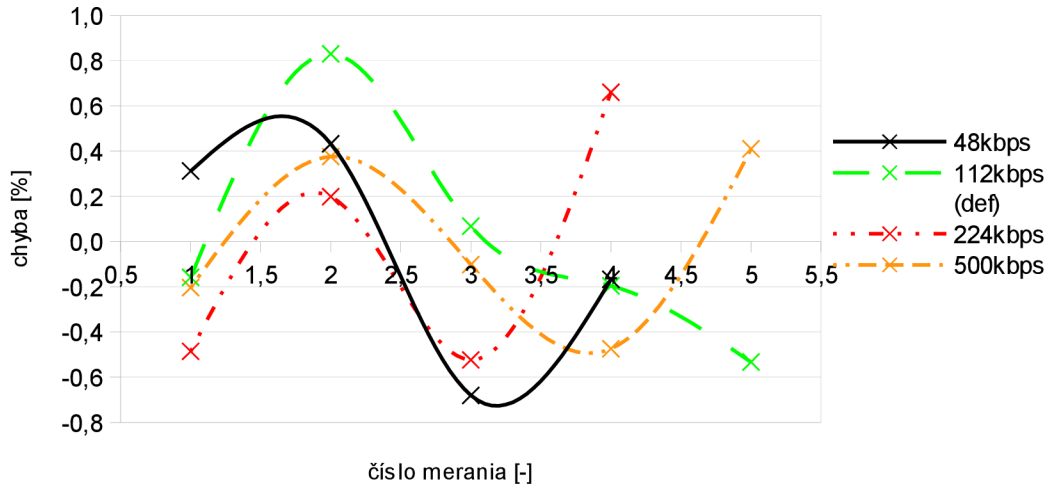
Obr. 5.5: Odchýlky merania kodeku ffmpeg

### Odchýlky merania pre LAME



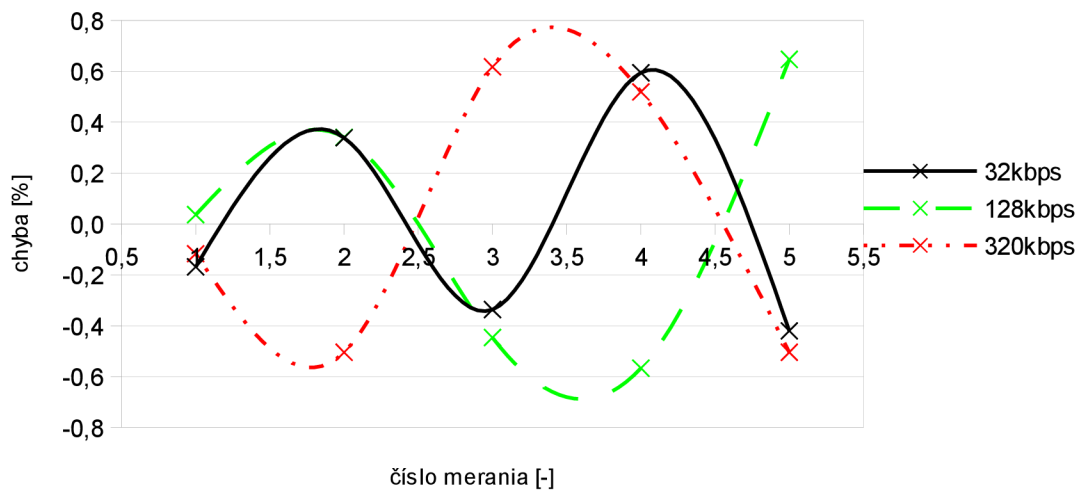
Obr. 5.6: Odchýlky merania kodeku lame

### Odchýlky merania pre Vorbis



Obr. 5.7: Odchýlky merania kodeku vorbis

### Odchýlky merania pre OPUS



Obr. 5.8: Odchýlky merania kodeku OPUS

## 5.2 PESQ

Objektívne merania kvality sa využívajú napríklad pri návrhu nového kodeku, kedy je potrebné zistiť kvalitu algoritmu.

Pre zistenie objektívnej kvality rečových nahrávok boli použité materiály od ITU (ITU-T P.862 (02/2001)).

Meranie som vykonal na nahrávkach dodaných spolu so zdrojovými kódmi z webu ITU. Pre spustenie bolo potrebné preložiť všetky zdrojové súbory pomocou

```
gcc -Wall dsp.c pesqdsp.c pesqio.c pesqmain.c pesqmod.c -lm -o pesq
```

kedy bol vytvorený spúšťač súbor *pesq*.

Po spustení programu *pesq* sa zobrazila nápoveda. Porovnanie sa spúšťa pomocou

```
PESQ [options] ref deg [smos] [cond]
```

kde ako voľbu *options* je možné zvoliť vzorkovací kmitočet „+8000, +16000, +swap“. Ďalej je potrebné zadať originálnu a degradovanú nahrávku.

Na obr.5.9 je zobrazený príklad výstupu programu po zadaní vzorkovacieho kmitočtu 8 kHz. Výsledná hodnota MOS je rovná 3,775.

```
Reading reference file /home/michal/nahravky/or229.wav...done.
Reading degraded file /home/michal/vystupy_kodekov/opus/dg229.wav...done.
Level normalization...
IRS filtering...
Variable delay compensation...
Acoustic model processing...

Prediction : PESQ_MOS = 3.775
michal@c03-630b:~/pesq$ █
```

Obr. 5.9: Ukážka výstupu referenčnej implementácie PESQ

Pre porovnanie kodekov som využil referenčné nahrávky pribalené v archíve PESQ. Jedná sa o *or105.wav*, *or129.wav*, *or179.wav*, *or229.wav*. Vytvoril som si aj vlastnú krátku nahrávku pod názvom *nazdar.wav* a jednu som prebral z [http://www.signallogic.com/melp/EngSamples/Orig/ENG\\_M.wav](http://www.signallogic.com/melp/EngSamples/Orig/ENG_M.wav).

Porovnával som kodeky G.711, G729 a OPUS, v ktorom som použil *voip* mód. Pri kódovaní pomocou opusu som nastavil dva rôzne bitrate – 8 a 6 kbps a pri G.711 som použil verzie  $\mu$  – *law* a *a* – *law*.

Výsledky merania sú zhrnuté v tab. 5.7.

Kvalita kodeku G.711 je podľa hodnotenia [33] výborná (5 – najlepšie skóre, 1 – najhoršie). Menšia odchýlka vznikla vo verzii *a* – *law* kde sa hodnota MOS nahrávky *nazdar* výrazne zmenšila na 3,528. Domnievam sa, že je to spôsobené dĺžkou nahrávky, ktorá je veľmi malá oproti zvyšným.

Tab. 5.7: Hodnoty MOS po prechode PESQ algoritmu

Nahrávka	or105	or129	or179	or229	nazdar	ENG_M
<b>G.711</b> <i>a-law</i>	4,5	4,499	4,5	4,5	3,528	4,347
<b>G.711</b> <i><math>\mu</math>-law</i>	4,5	4,5	4,5	4,5	4,5	4,5
<b>G.729</b>	3,756	3,909	3,825	3,958	3,18	3,597
<b>OPUS</b> <i>8 kbps</i>	3,817	3,928	3,729	3,893	3,263	3,654
<b>OPUS</b> <i>6 kbps</i>	3,693	3,816	3,559	3,751	3,148	3,409

Kodek G.729 podáva horšie výsledky. Dôvodom je použitie stratového kódovania a nižšieho bitrate. Podľa hodnotenia kvality MOS je kvalita ešte dobrá.

Kvalita kodeku OPUS vo *voip* móde je tiež dobrá. Rovnako ako u predchádzajúcich kodekov bol väčší rozdiel v kvalite pri nahrávke *nazdar*. Malým pokusom bolo nastavenie dvoch rozdielnych bitrate a to 8 a 6 kbps. So znižujúcim bitrate sa znížila aj kvalita. Kodér síce neurčuje minimálny bitrate, ale všimol som si, že pri nastavení na 3 a 1 kbps bol výsledok rovnaký.

## 6 ZÁVER

Témou tejto diplomovej práce sú zvukové a rečové kodeky. Mojou úlohou bolo porovnať voľne dostupné implementácie z hľadiska výpočtovej náročnosti a pokúsiť sa o ich porovnanie pomocou vhodnej objektívnej metódy.

V kapitolách boli popísané základné vlastnosti ľudského sluchu, psychoakustický model, ktorým sa riadi stratová kompresia, kódovanie zvuku a reči. Pri kódovaní zvuku sú popísané rozdiely medzi stratovým a bezstratovým kódovaním a boli vybraní predstavitelia – pre bezstratové kódovanie FLAC a pre stratové MPEG Layer 3 a OGG Vorbis. Kódovanie reči zahŕňa popis lineárneho predikčného kódovania a lineárnej predikcie, ktorá bola prakticky overená, v programe Matlab. Kapitola o rečových kodekoch obsahuje aj popis kodekov G.729 a OPUS. Objektívnym metódam bola venovaná ďalšia kapitola a poslednou kapitolou je praktické porovnanie výpočtovej náročnosti jednotlivých kodekov a porovnanie rečových kodekov na základe objektívnej miery – PESQ.

Vykonaním testov pamätovej náročnosti som zistil špičky využitia pamäte pri kódovaní skladieb. Túto informáciu by som mohol ďalej využiť pri návrhu prenosného prehrávača alebo iného zariadenia, kde je kladený dôraz na výpočtový výkon. Kodér ffmpeg vyšiel z tohto testu ako najmenej vhodný pre použitie v zariadení s obmedzenou operačnou pamäťou, pretože oproti ostatným kodérom využíval najviac pamäte, až 1,98 MB, kým lame len približne 450 kB pri rôznych nastaveniach kvality. Kodér vorbisu sa ukázal pri veľmi nízkom bitrate tiež ako kodér náročný na pamäť. Dôvodom je obsiahla kódová kniha.

Ako najmenej časovo náročný kodek sa ukázal FLAC, pretože kodér vykonáva iba Huffmanovo alebo LZ kódovanie. So vzrastajúcou kompresiou vzrastal aj výsledný čas kódovania (najnižšia kompresia 0,9432 s, najvyššia 7,2432 s). Väčší časový rozdiel kódovania v nižšej a vyššej kvalite som si všimol u kodeku lame, kedy kódovanie skladby na 32kbps bitrate trvalo necelých 7 sekúnd, na 128 kbps a vyššie necelých 13 sekúnd. Rozdiel medzi 128 a 320 kbps už nebol veľký, iba približne 100 ms. Kodér ffmpeg sa oproti lame vyznačuje vyššou časovou spotrebou pri nízkom bitrate (približne 1,5 s) a nižšou pri vyššom bitrate (približne 3 s). Kodér vorbisu vykazoval malé časové rozdiely pri rôznych kvalitách, pri najvyššom bitrate sa výpočtový čas zvýšil o približne 1,25 s. Kodek opus sa od všetkých ostatných spomínaných líši tým, že so vzrastajúcim bitrate vzrastá v celom rozsahu aj výpočtový čas.

Z toho vyplýva, že kodér lame zrejme obsahuje časti, ktoré sa pri nízkom bitrate obchádzajú. Výsledky merania času považujem za vierohodné, pretože odchýlky od priemeru nečinili okrem jedného prípadu pri FLAC kodeku ani jedno percento.

Výsledkom objektívneho porovnania rečových kodekov som zistil, že pre použitie v IP telefónii sú vhodné všetky. Najlepšie výsledky podával G.711, kedy sa hodnota

MOS pohybovala na hranici 4,5, pretože ide o bezstratové kódovanie. Kodeky G.729 a OPUS poskytujú nižšiu kvalitu, naopak na úkor šetrenia dátového toku. V prípade neobmedzených sieťových možností je vhodné použiť G.711 a v prípade, že nám to kapacity nedovoľujú, zvyšné dva kodeky.

# LITERATÚRA

- [1] BENEŠ, Miroslav. *Huffmanovo kódování* [online]. [cit. 2011-11-25]. Huffmanovo kódování. Dostupné z WWW: <<http://www.cs.vsb.cz/benes/vyuka/pte/texty/kompresse/ch02s02.html>>.
- [2] BRANDENBURG, K. *MP3 and AAC explained*. [online]. [cit. 2011-12-4]. Dostupný z WWW: <[http://www.telos-systems.com/techtalk/hosted/Brandenburg\\_mp3\\_aac.pdf](http://www.telos-systems.com/techtalk/hosted/Brandenburg_mp3_aac.pdf)>.
- [3] C. CHU, Wai. *Speech Coding Algorithms : Foundation and Evolution of Standardized Coders*. San Jose, California : John Wiley & Sons, 2003. 578 s.
- [4] C. LOIZOU, Philipos. *Speech enhancement : Theory and Practice*. Canada : CRC Press, 2007. 608 s.
- [5] GRUNDLEHNER, B., et al. *Performance assessment method for speech enhancement systems*. [online]. [cit. 2011-12-1]. Dostupný z WWW: <[http://www2.math.umd.edu/~rvbalan/PAPERS/MyPapers/sps2005\\_speech\\_final.pdf](http://www2.math.umd.edu/~rvbalan/PAPERS/MyPapers/sps2005_speech_final.pdf)>.
- [6] CHEN, Howard; L. YU, T. *Comparison of Psychoacoustic Principles and Genetic Algorithms in Audio Compression*. [online]. 27 december 2005, [cit. 2011-11-26]. Dostupný z WWW: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1562864>>.
- [7] JEŽEK, David. *Informace a novinky o hardware, software a internetu - DIIT* [online]. 1.11.2011 [cit. 2011-11-12]. Apple uvolnil bezztrátový zvukový kodek ALAC. Dostupné z WWW: <<http://www.diit.cz/clanek/apple-uvolnil-bezztratovy-zvukovy-kodek-alac/39013/>>.
- [8] KOSAKA, Atsushi, et al. *VLSI Implementation of Ogg Vorbis Decoder for Embedded Applications*. [online]. 06 January 2003 , [cit. 2011-11-28]. Dostupný z WWW: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1158024>>.
- [9] NEUHOFF, David. *Design of a CELP coder and analysis of various quantization techniques* . Michigan, Winter 2005. 27 s. EECS 651 Project Report. University of Michigan, Ann Arbor.
- [10] NOLL, Peter. *Wideband Speech and Audio Coding*. IEEE Communications Magazine. 1993, s. 34-43.



- [11] Ó CINNÉIDE, Alan. *Linear Prediction The Technique, Its Solution and Application to Speech*. [online]. August 2008, [cit. 2011-11-27]. Dostupný z WWW: <<http://eleceng.dit.ie/papers/92.pdf>>.
- [12] PARSA, Vijay; JAMIESON, Donald. *Evaluation of the G.729 speech coder with pathological voice samples*. [online]. 2001, [cit. 2011-12-2]. Dostupný z WWW: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=933740>>.
- [13] PELANT, Lukáš. *Maskování akustických signálů*. Praha, 22.10.2007. 13 s. Semestrální práce. České Vysoké Učení Technické v Praze, Fakulta elektrotechnická. Dostupné z WWW: <[http://dsp.vscht.cz/konference\\_matlab/MATLAB07/prispevky/pelant\\_1/pelant\\_1.pdf](http://dsp.vscht.cz/konference_matlab/MATLAB07/prispevky/pelant_1/pelant_1.pdf)>.
- [14] PHAMDO, Nam. *Lossless Data Compression* [online]. 2000 [cit. 2011-11-22]. Lossless Data Compression. Dostupné z WWW: <<http://www.data-compression.com/lossless.shtml>>.
- [15] RABHA, Pankaj. *Scalable Wideband Coding of Speech*. [online]. [cit. 2011-12-3]. Dostupný z WWW: <<http://www.stanford.edu/~raghuram/wideband.pdf>>.
- [16] RICHARDS, D. *Speech transmission performance of PCM systems*, Electron. Lett. 1, 1965, 40-41
- [17] SAYOOD, Khalid. *Introduction to Data Compression*, 3rd ed.. San Francisco : Elsevier Inc., 2006. 680 s.
- [18] SOVIČ, Dušan. *Huffmanovo kódovanie* [online]. 2000, 4. Októbra 2005 [cit. 2011-11-25]. Huffmanovo kódovanie. Dostupné z WWW: <<http://www.pakuj.host.sk/single/huffman.html>>.
- [19] ŠVEC, Jiří. *Komprese zvuku? Jen podvod na uši! - AVmania.cz* [online]. 10.6.2009 [cit. 2011-12-12]. Komprese zvuku? Jen podvod na uši!. Dostupné z WWW: <<http://avmania.e15.cz/komprese-zvuku-jen-podvod-na-usi>>.
- [20] TESAŘ, P. *Vysílání multimediálního obsahu s využití kompresních technik*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 80 s. Vedoucí diplomové práce Ing. Jakub Müller.
- [21] Tweak. *16 vs. 24 bit Audio Recording Demystified* [online]. 2010 [cit. 2011-11-12]. 16 Bit vs. 24 Bit Audio. Dostupné z WWW: <[http://tweakheadz.com/16\\_vs\\_24\\_bit\\_audio.htm](http://tweakheadz.com/16_vs_24_bit_audio.htm)>.

- [22] WICHMAN, Shannon. *A Comparison of Speech Coding Algorithms ADPCM vs CELP*. [online]. December 8, 1999, [cit. 2011-12-11]. Dostupný z WWW: <<http://www.utdallas.edu/~torlak/wireless/projects/wichman.pdf>>
- [23] Xiph.Org Foundation. *Vorbis I specification* [online]. 1994, August 11, 2011 [cit. 2011-11-23]. Vorbis I specification. Dostupné z WWW: <[http://xiph.org/vorbis/doc/Vorbis\\_I\\_spec.html](http://xiph.org/vorbis/doc/Vorbis_I_spec.html)>.
- [24] *Audio compression (data)#Lossless audio compression*. [online]. 22 October 2001, last modified on 9 December 2011 [cit. 2011-12-12]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Audio\\_compression\\_\(data\)#Lossless\\_audio\\_compression](http://en.wikipedia.org/wiki/Audio_compression_(data)#Lossless_audio_compression)>.
- [25] *Bark - Hydrogenaudio Knowledgebase* [online]. , 17 February 2007 [cit. 2011-12-3]. Bark. Dostupné z WWW: <<http://wiki.hydrogenaudio.org/index.php?title=Bark>>.
- [26] *Critical band - Hydrogenaudio Knowledgebase* [online]., 11 September 2006 [cit. 2011-12-3]. Critical band. Dostupné z WWW: <[http://wiki.hydrogenaudio.org/index.php?title=Critical\\_band](http://wiki.hydrogenaudio.org/index.php?title=Critical_band)>.
- [27] *Číslíkové zpracování akustických signálů: Studijní materiály ke zkoušce* [online]. Brno, 2010, 8.12.2010 [cit. 2012-05-15]. Dostupné z: <[https://www.vutbr.cz/www\\_base/priloha.php?dpid=40695](https://www.vutbr.cz/www_base/priloha.php?dpid=40695)>. Skriptum. Vysoké učení technické v Brně. Vedoucí práce Ing. Miroslav Balík, Ph. D.
- [28] Draft-ietf-codec-opus-11. *Definition of the Opus Audio Codec*. 12. vyd. Mountain View, CA: Network Working Group, 2012. Dostupné z: <<http://tools.ietf.org/html/draft-ietf-codec-opus-12>>.
- [29] Draft-vos-silk-02. *SILK Speech Codec*. Stockholm: Network Working Group, 2011. Dostupné z: <<https://developer.skype.com/silk/draft-vos-silk-02.txt>>.
- [30] *FLAC - Wikipedia, the free encyclopedia* [online]. 2001, 6.12.2011 [cit. 2011-12-8]. FLAC. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Free\\_Lossless\\_Audio\\_Codec](http://en.wikipedia.org/wiki/Free_Lossless_Audio_Codec)>.
- [31] *Free Lossless Audio Codec - Hydrogenaudio Knowledgebase. Free Lossless Audio Codec* [online]. 27 July 2011 [cit. 2012-05-15]. Dostupné z: <[http://wiki.hydrogenaudio.org/index.php?title=Free\\_Lossless\\_Audio\\_Codec](http://wiki.hydrogenaudio.org/index.php?title=Free_Lossless_Audio_Codec)>.
- [32] *Ogg*. [online]. 29 September 2001, last modified on 16 November 2011 [cit. 2011-12-12]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Ogg>>.

- [33] *Quality, Voice Quality, MOS, NIQA, PESQ, Network Monitor, GSM, ITU.* Voice Quality [online]. 2010 [cit. 2012-05-12]. Dostupné z WWW: <[http://arptel.com/products\\_vq\\_voice\\_quality.html](http://arptel.com/products_vq_voice_quality.html)>.

## ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

ALAC	Apple Lossless Audio Codec
CBR	Constant Bitrate – Konštantný dátový tok
CELP	Code Excited Linear Prediction
CELT	Constrained Energy Lapped Transform – Transformácia s obmedzenou energiou
D/A	Digital to Analog – Digitálne analógový
dB SPL	Decibel, Sound Pressure Level – Úroveň akustického tlaku
DTX	Discontinuous Transmission – Nesúvislý prenos
FEC	Forward Error Correction – Dopredná korekcia chýb
FFT	Fast Fourier transform – Rýchla Fourierova transformácia
FLAC	Free Lossless Audio Codec
$f_{vz}$	Vzorkovací kmitočet
LP	Lineárna Predikcia
LPC	Linear Predictive Coding – Lineárne predikčné kódovanie
LSP	Line Spectrum Pair
LZ	Lempel-Ziv
MDCT	Modified discrete cosine transform – Modifikovaná diskretná kosínová transformácia
MOS	Mean opinion score
MP3	MPEG 1/2 Layer III
OS	Operačný systém
PESQ	Perceptual Evaluation of Speech Quality – Objektívne hodnotenie kvality reči
RMS	Root mean square
SNR	Signal-to-noise ratio – Odstup signál - šum

TOC	Table Of Contents – tabuľka obsahu
VBR	Variable Bitrate – Premennivý dátový tok
WAVE	Waveform Audio File Format

# ZOZNAM PRÍLOH

<b>A Príloha 1</b>	<b>50</b>
A.1 Implementácia lineárnej predikcie . . . . .	50
<b>B Príloha 2</b>	<b>52</b>
B.1 Grafický prehľad pamätovej náročnosti . . . . .	52
<b>C Príloha 3</b>	<b>61</b>
C.1 Obsah CD . . . . .	61

# A PRÍLOHA 1

## A.1 Implementácia lineárnej predikcie

```
clear all
close all
clc

%nacitanie wav
[wav fs] = wavread('priklad.wav');

%urcenie pomocnej matice ktora bude sluzit na doplnenie posledneho segmentu
dlzka = floor(length(wav)/64);
a = zeros((dlzka+1)*64-length(wav),1);

%doplnenie matice o 'a' aby vznikli pevne segmenty
novywav = [wav',a']';

%rozdelenie na segmenty (kazdy riadok = 1 segment, 1 segment = 64 vzoriek)
segmentyL = reshape(novywav,64,dlzka+1);

koef = 0;
lag=15;
for g = 1:dlzka
    %vyber jedneho segmentu
    Segment = segmentyL(:,g);

    %vypocet autokorelacie a toeplitzovej matice
    [autokorelacia]=autokor(Segment,lag);

    %vypocet koeficientov "a"
    a = -autokorelacia\autokorelacia(:,1);
    koef = [koef a'];
end

%ulozenie koeficientov do vyslednej matice
%kazdy stlpec = koeficienty pre jeden segment
```

```
LPkoef = reshape(koef(2:end),length(a),dlzka);
```

Funkcia autokor:

```
function autokorelacia = autokor(Segment,lag)
    for i=0:lag-1
        if i==0
            l(i+1)=sum(Segment*Segment(1));
        else
            h=Segment(1:i:end);
            l(i+1)=sum(sum(Segment(1:length(h))*h'));
        end
    end
    autokorelacia=toeplitz(l);
end
```



## B PRÍLOHA 2

### B.1 Grafický prehľad pamäťovej náročnosti

Na obr. B.1 je špička využitia pamäte 1,98 MB.

Na obr. B.2 je špička využitia pamäte 1,98 MB.

Na obr. B.3 je špička využitia pamäte 457 kB.

Na obr. B.4 je špička využitia pamäte 437 kB.

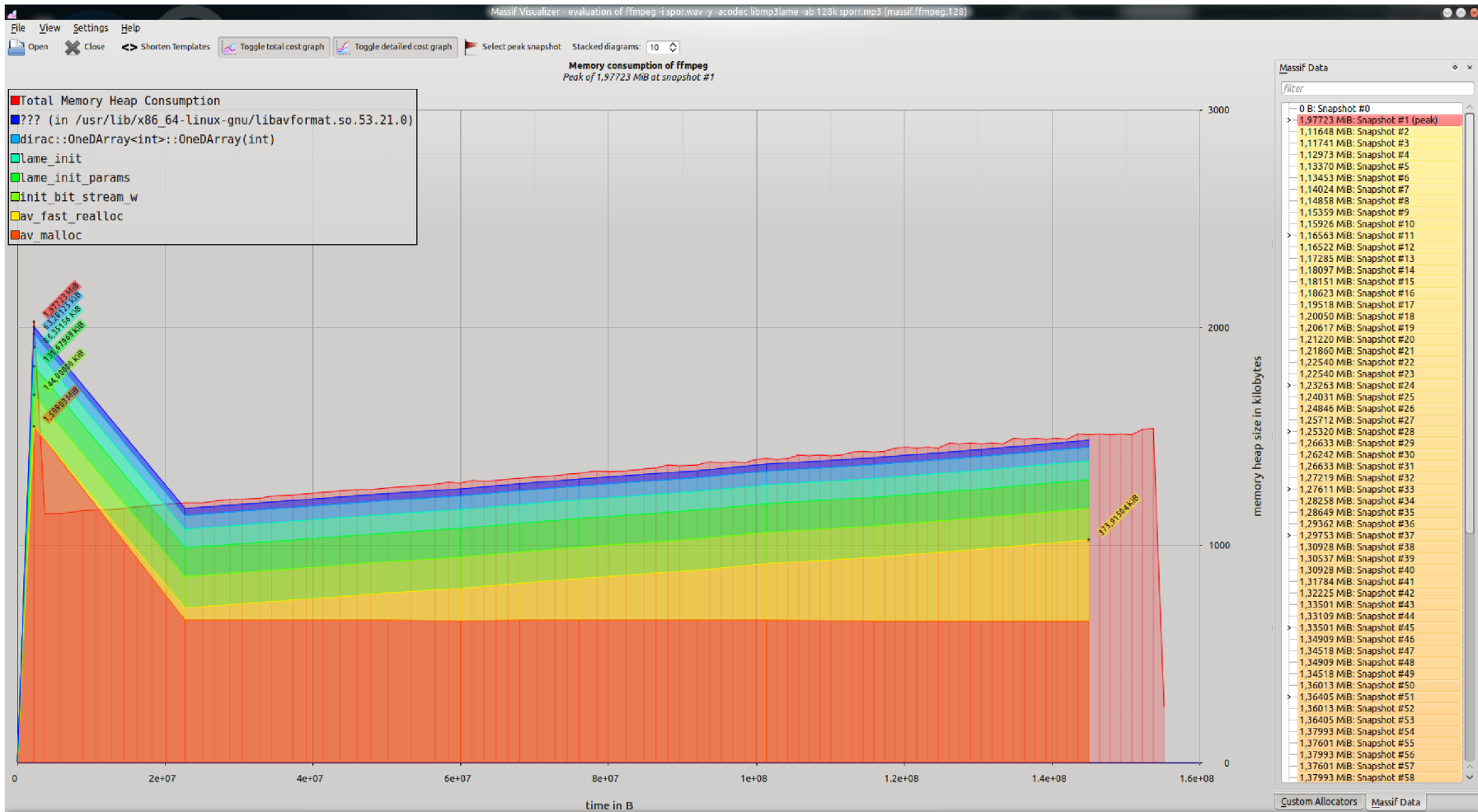
Na obr. B.5 je špička využitia pamäte 437 kB.

Na obr. B.6 je špička využitia pamäte 1,02 MB.

Na obr. B.7 je špička využitia pamäte 525,9 kB.

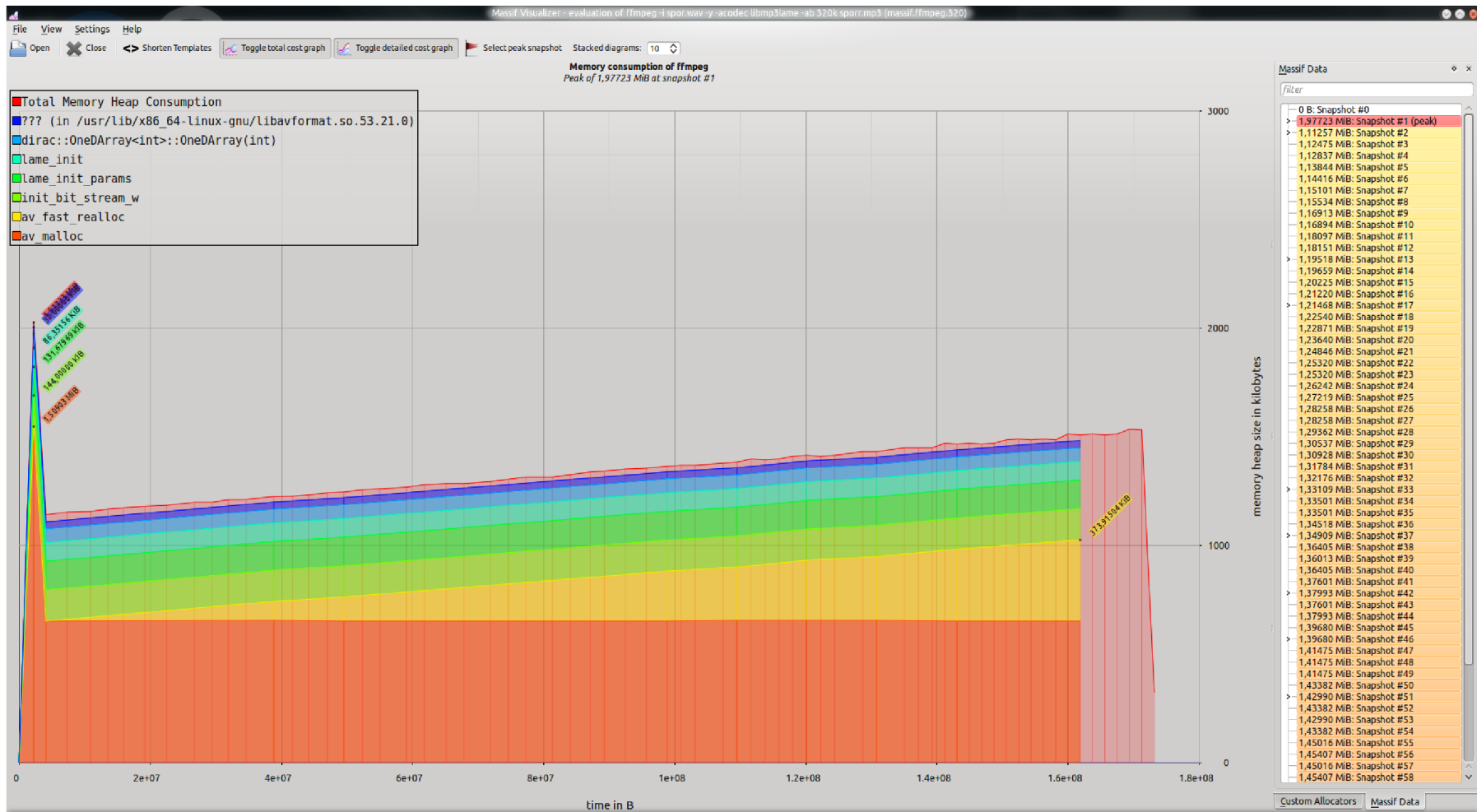
Na obr. B.8 je špička využitia pamäte 505,5 kB.

54

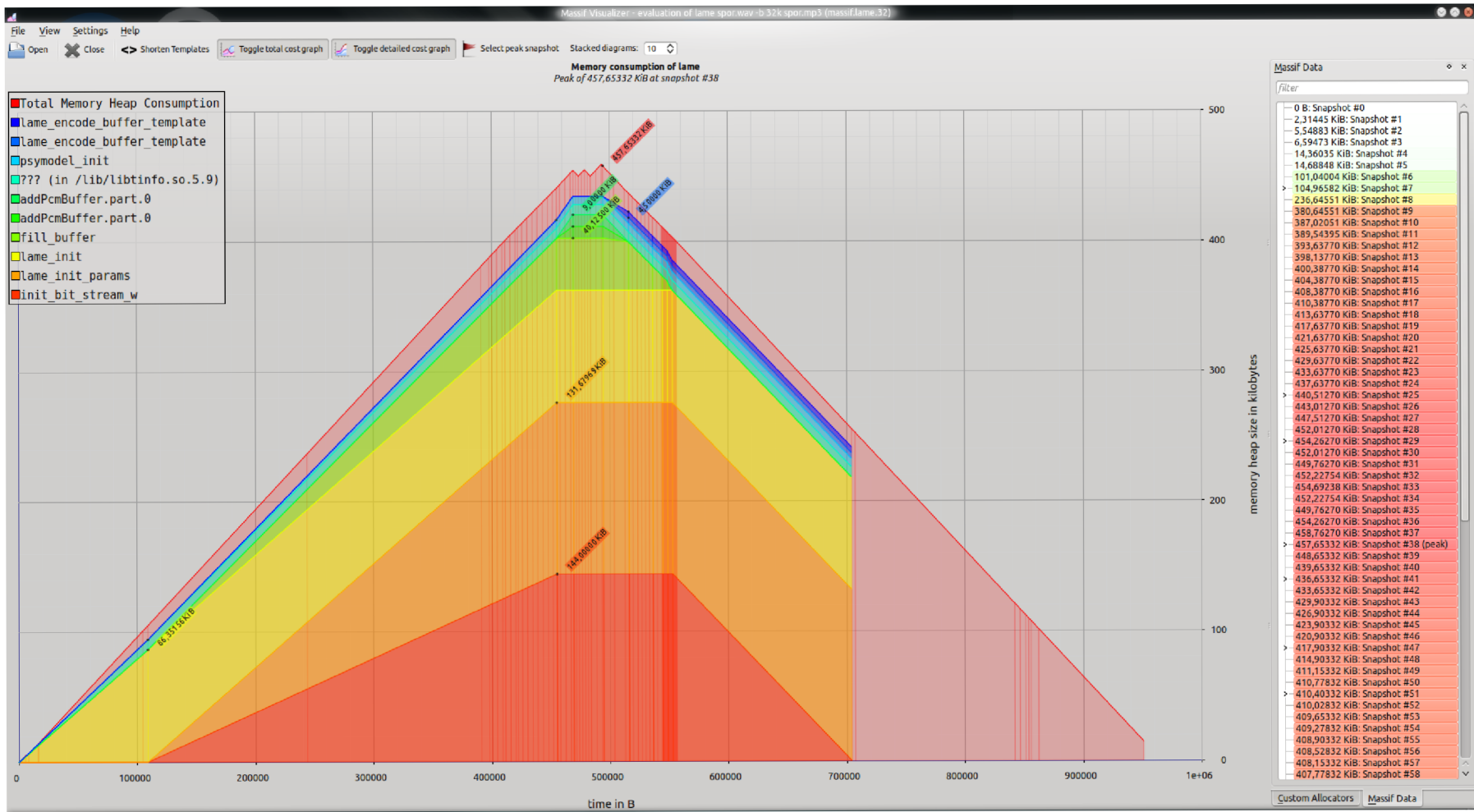


Obr. B.1: FFMPEG 128 kbps

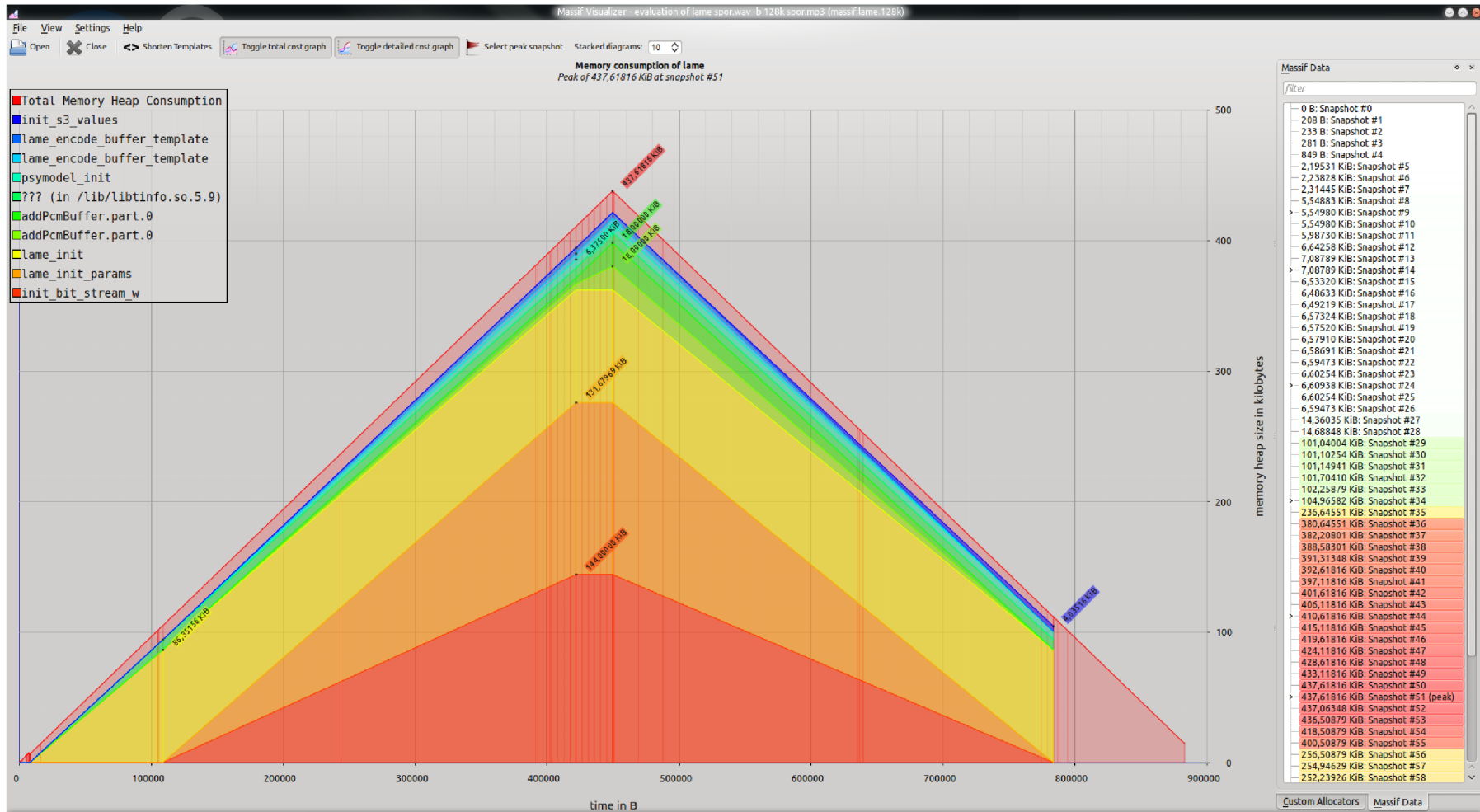
CG



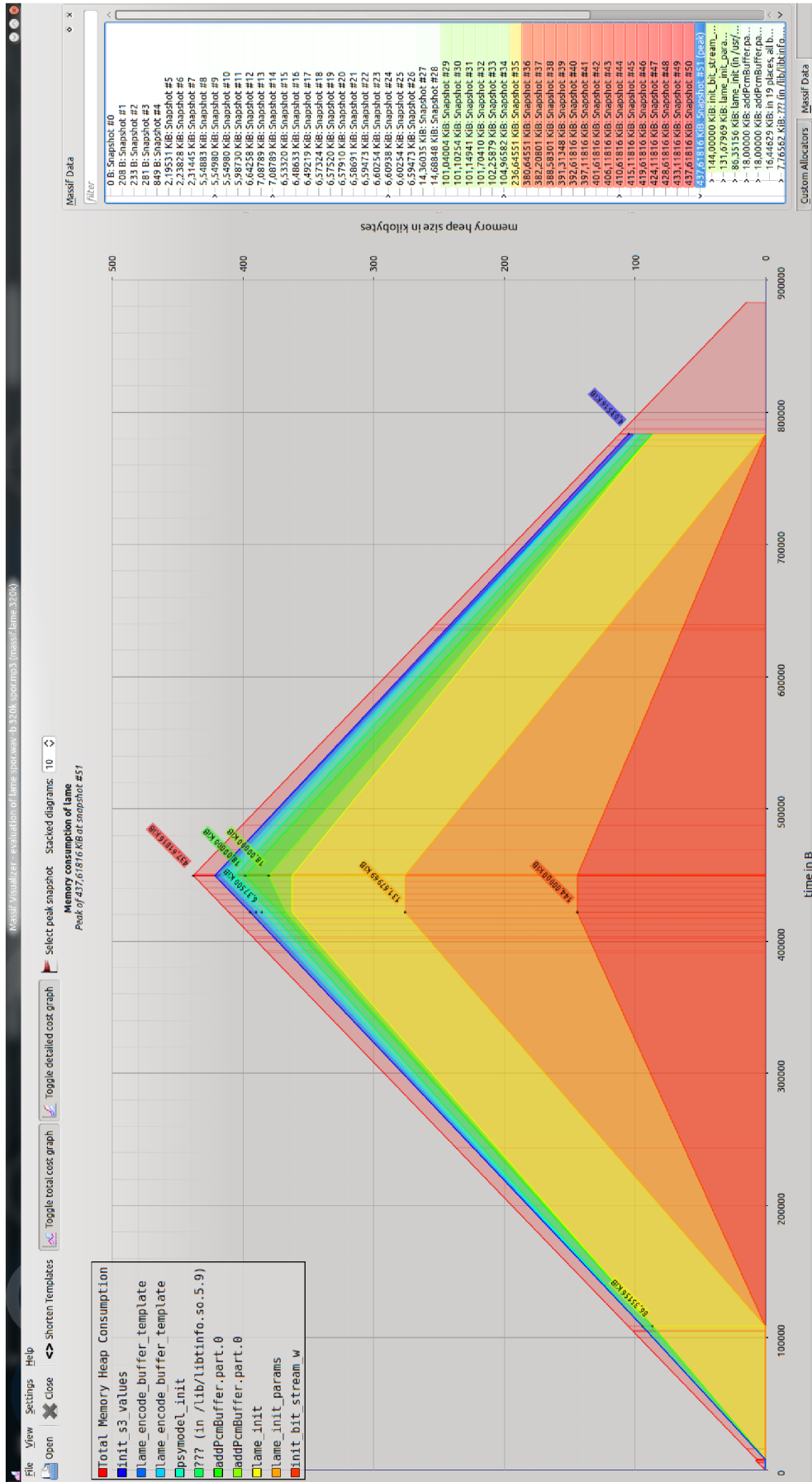
Obr. B.2: FFMPEG 320 kbps



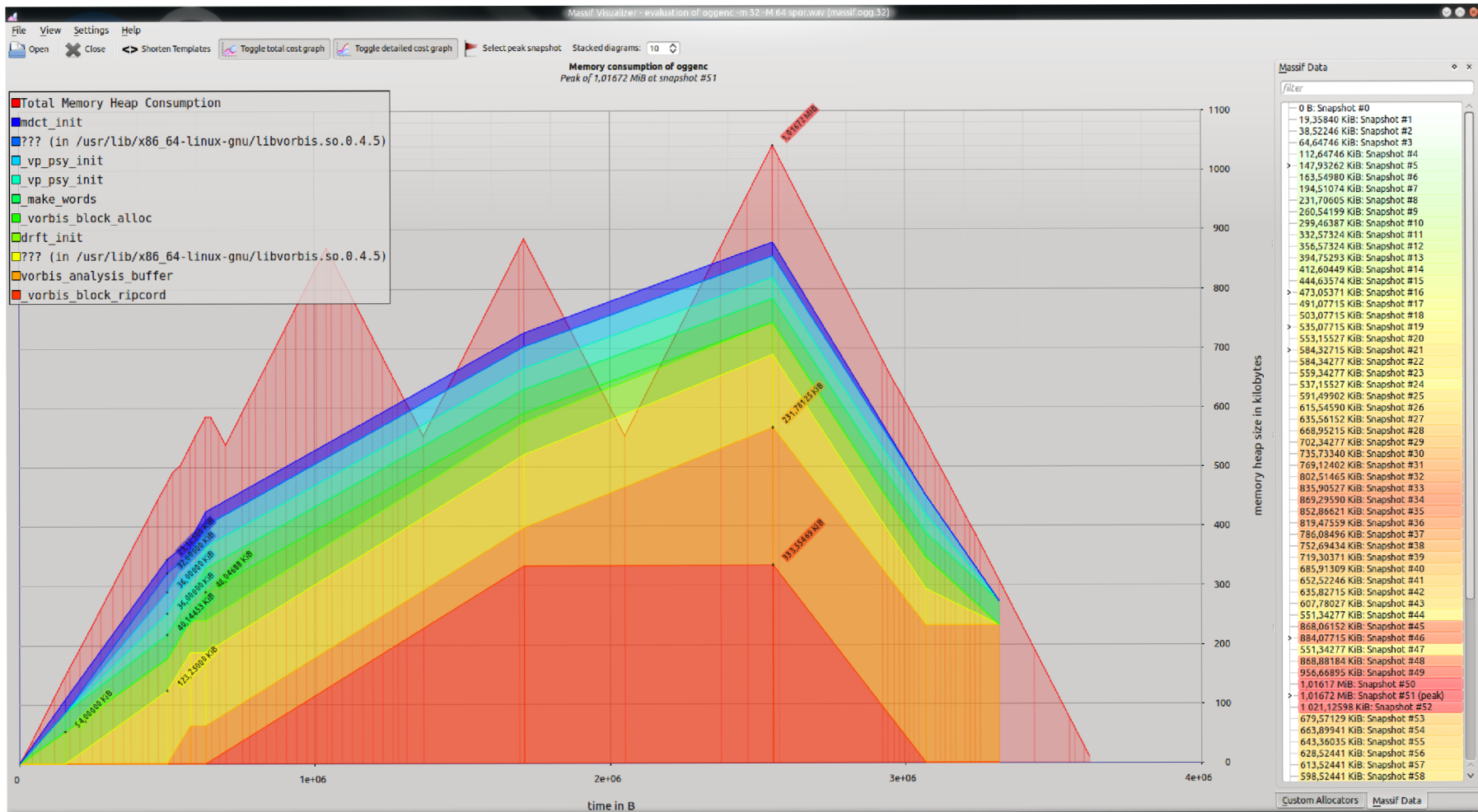
Obr. B.3: LAME 32 kbps



Obr. B.4: LAME 128 kbps

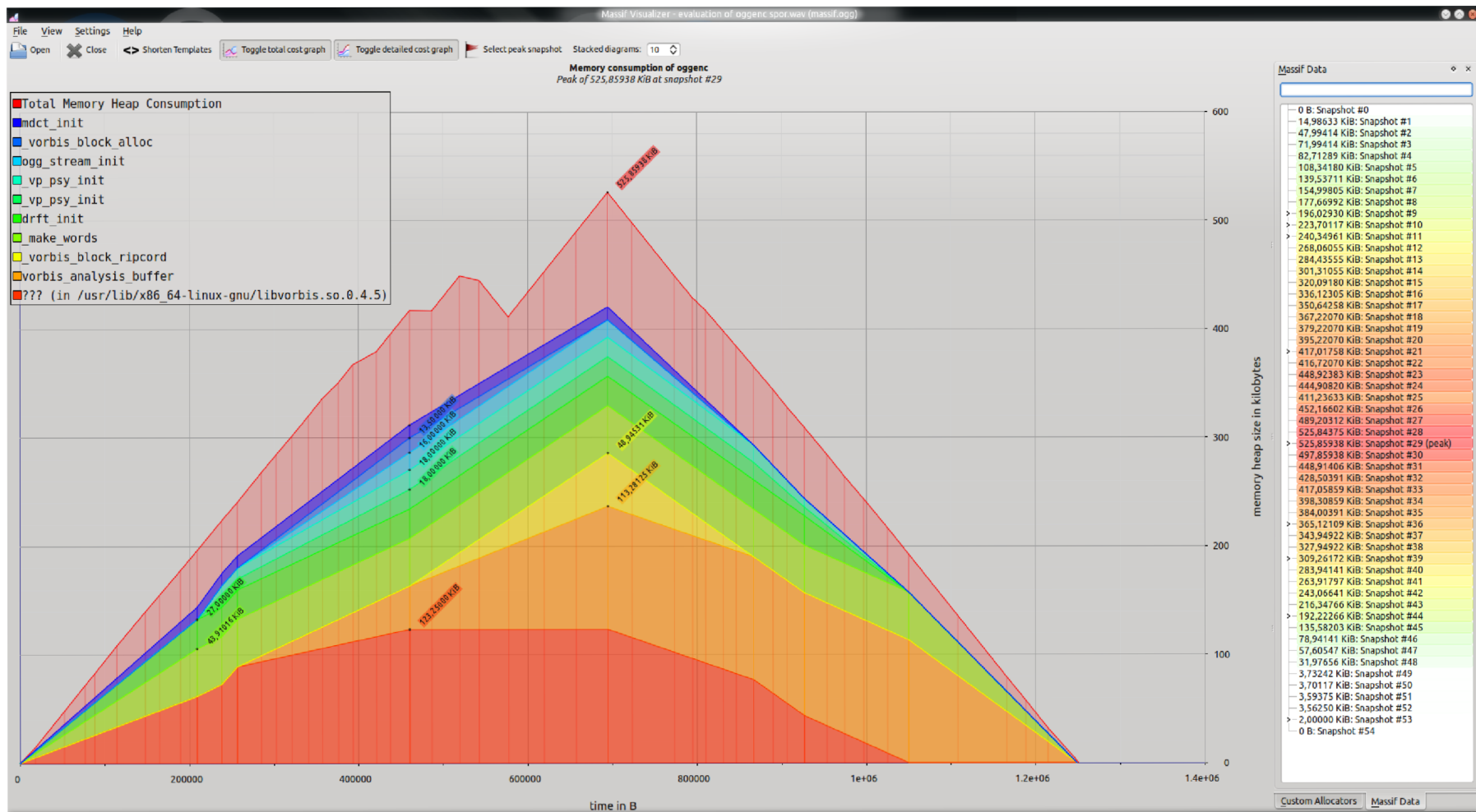


Obr. B.5: LAME 320 kbps



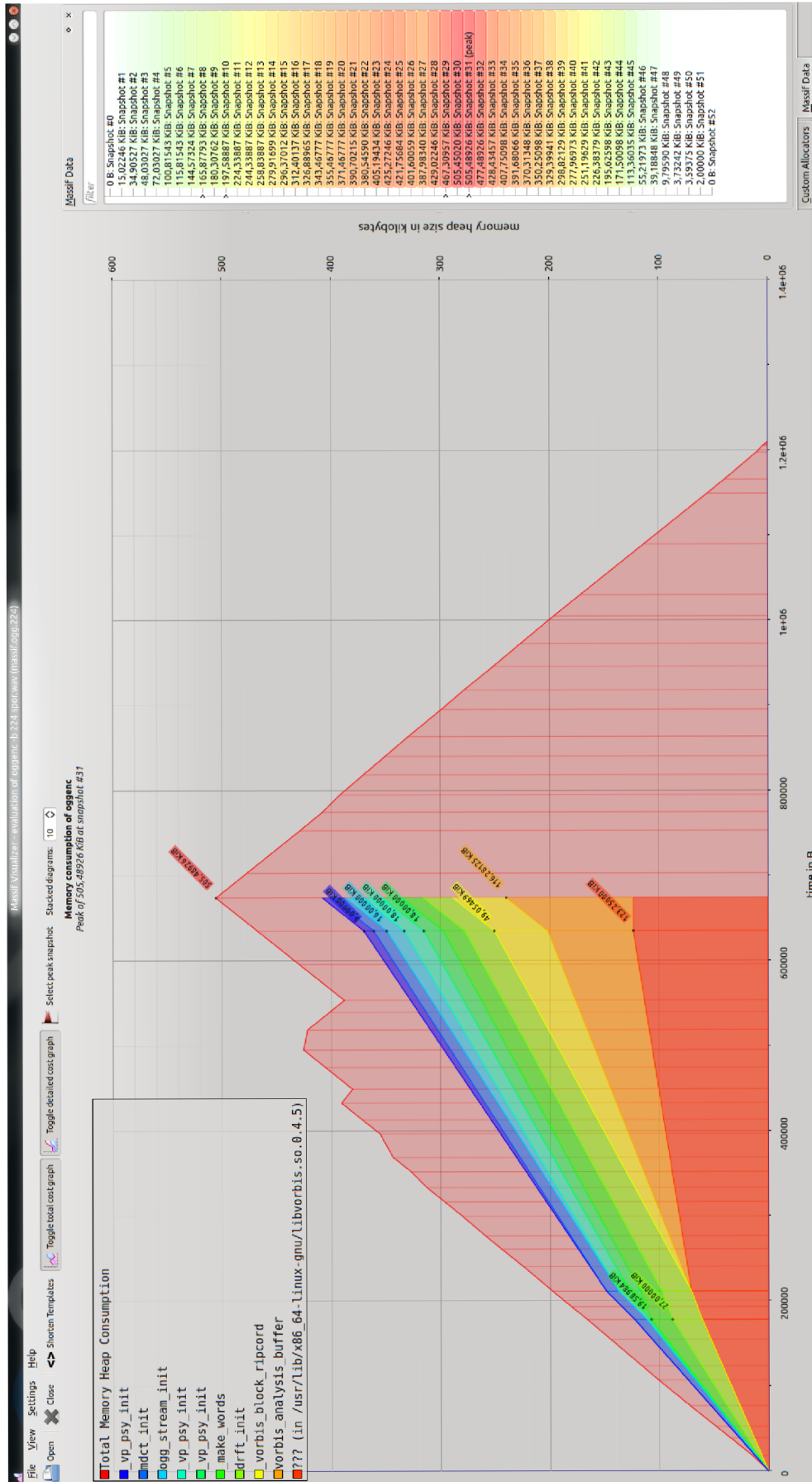
Obr. B.6: Vorbis 32-64 kbps





Obr. B.7: Vorbis 112 kbps





Obr. B.8: Vorbis 224 kbps

## C PRÍLOHA 3

### C.1 Obsah CD

-> g711 -> alaw -> dg105, dg105.wav, dg129, dg129.wav, dg179,  
dg179.wav, dg229, dg229.wav, eng, eng.wav,  
nazdar, nazdar.wav  
-> ulaw -> dg105, dg105.wav, dg129, dg129.wav, dg179,  
dg179.wav, dg229, dg229.wav, eng, eng.wav,  
nazdar, nazdar.wav  
-> g729 -> dg105, dg105.wav, dg129, dg129.wav, dg179, dg179.wav,  
dg229, dg229.wav, eng, eng.wav, nazdar, nazdar.wav  
-> opus -> dg105.wav, dg105-6k.wav, dg129.wav, dg129-6k.wav,  
dg179.wav, dg179-6k.wav, dg229.wav, dg229-6k.wav,  
dg-nahravka.wav, dg-nahravka-6k.wav, eng-6k.wav,  
nazdar-6k.wav, nazdar-8k.wav  
-> referencne -> ENG\_M.wav, nazdar.wav, or105.wav, or129.wav,  
or179.wav, or229.wav  
readme.txt