# Kódování a komprese – dokumentace gif2bmp

Bc. Jaroslav Sendler, xsendl00 xsendl00@stud.fit.vutbr.cz

8. března 2012

# Knihovna

Knihovna se stará o konverzi obrázku GIF na formát BMP. Jako vstup je akceptován jak verze gifu GIF89a tak i starší formát GIF87a. Výstupem je obrázek ve formátu BMP verze 3.

### **GIF**

Standard GIF89a obsahuje několik povinných položek, které jsou společné i pro starší verzi GIF87a. Pro tyto i další rozšiřující bloky jsou vytvořeny struktury reprezentující je. Při převodu na formát BMP jsou rozšiřujícími bloky naplněny odpovídající struktury, ale dále jsou ignorovány. Knihovna si poradí i s prokládanými obrázky.

Pro dekompresi obrázku komprimovaného pomocí algoritmu LZW byl použit kód popsaný pseudokódem 1. Pseudokód neobsahuje kontrolu přetečení velikosti slovníku a zvětšování počtu načítaných bitů. Počáteční velikost slovníku se nastaví podle následujícího výpočtu: velikost slovníku = počet barev « 1. Při naplnění slovníku se provede dříve uvedený výpočet a zvýší se počet načítaných bitů o jeden až do velikosti 12. Při situaci, kdy by se mělo načítat 13 bitů je počet snížen na 12 a následně je očekáván příchod hodnoty CC, kód značící reinicializaci slovníku. Po tomto kroku se vše nastaví na počáteční hodnoty a pokračuje se od začátku až do příchodu indexu kódu EOI.

```
inicializace slovníku
1
     načti první kódové slovo CODE ze vstupu
                                                            /* CODE index do slovníku */
3
     výstup += {CODE}
                                                            /* {CODE} obsah slovníku na daném indexu */
     while (vstup není prázdný) {
5
         načti další kódové slovo CODE ze vstupu
6
         if(CODE je ve slovníku) {
            výstup += {CODE}
K je první index z {CODE}
7
8
9
            přidej {CODE - 1} + K do slovníku
10
                                                            /* CODE není ve slovníku */
11
         else
12
            K je první index z {CODE - 1}
            v\acute{y}stup += {CODE - 1} + K
přidej {CODE - 1} + K do slovníku
13
14
15
         }
16
     }
```

Pseudokód 1: LZW algoritmus - dekomprese

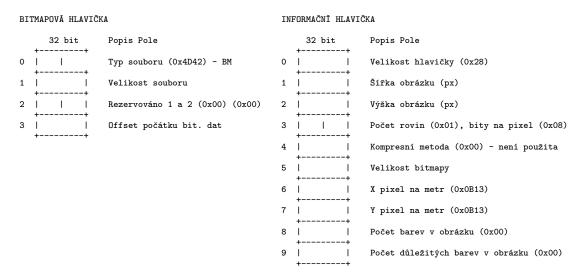
Jako rozhraní pro použití knihovny v aplikacích je k dispozici následující funkce int gif2bmp(tGIF2BMP \*gif2bmp, FILE \*inputFile, FILE \*outputFile), vracející 0 při úspěšném převodu nebo -1 při chybě. V případě jakékoliv závažné chybě, při které převod nemůže proběhnout je na standardní chybový výstup vypsán informační log a aplikace je ukončena

# BMP - Microsoft Windows Bitmap

Jako výstup je použit obrázek typu BMP, je použita verze 3 (Microsoft Windows 3.x). Verze 3 bitmapy obsahuje tři základní povinné sekce : bitmapovou hlavičku, informační hlavičku a bitmapu. Další část může být paleta, jež

je volitelná. Tato knihovna využívá pouze povinné sekce, které jsou implementovány strukturami tBmpHeader a tBmpInfoHeader. Jelikož je použito 24 bitů na pixel, není využita barevná paleta, jelikož pixelová barevná data jsou uložena přímo v obrazových datech. Samotná obrazová data nejsou komprimována žádným algoritmem. Tedy výstupní soubor je mnohem větší než vstupní obrázek GIF komprimovaný LZW algoritmem.

Níže uvedené schéma 1 reprezentuje použité struktury v obrázku. První je Bitmapová hlavička, která je následována Informační hlavičkou. Zbytek souboru tvoří bitmapová data. V závorkách na koncích popisu schématu jsou uvedena data (neměnná), jež jsou použita u každého obrázku generovaného touto knihovnou.



Obrázek 1: Přehled hlaviček obrázku formátu BMP verze 3.

# **Aplikace**

Aplikace je konzolového charakteru, s uživatelem komunikuje pomocí počátečních vstupních parametrů příkazové řádky uvedených v tabulce 1.

přepínač	parametr	popis
-i	<ifile $>$	vstupní soubor obrázku, pokud parametr není vstup je STDIN
-O	<ofile $>$	výstupní soubor obrázku, pokud parametr není výstup je STDOUT
-l	<logfile $>$	vypíše informace o loginu, velikosti vstupního obrázku a výstupního
-h		vypíše nápovědu na standardní výstup, ukončí aplikaci

Tabulka 1: Přehled vstupních parametrů příkazové řádku.

Pro zpracování vstupních parametrů byla použita funkce getop.