

# Kódování a komprese dat 2015/2016: Projekt č. 3

**Název:** Konverze obrazového formátu GIF na BMP

**Datum odevzdání:** nejpozději do 08. 05. 2016 prostřednictvím IS FITu.

**Forma odevzdání:** elektronicky - soubory v archívu ZIP

**Počet bodů:** max. 30

**Poznámka:** pro účely testování lze využít např. logo FITu z webových stránek

**Dotazy:** simekv@fit.vutbr.cz

## **Zadání:**

V jazyce C/C++ naimplementuje knihovnu a aplikaci pro převod souboru grafického formátu GIF (Graphics Interchange Format) na soubor grafického formátu BMP (Microsoft Windows Bitmap). Pro soubor ve formátu GIF se předpokládá jeho zakódování pomocí metody LZW. Výstupní soubor ve formátu BMP pak musí být možné zobrazit např. prohlížečem Irfan View či jiným vhodným.

## **Podrobné pokyny k vypracování:**

### **a) implementace knihovny:**

V rozhraní knihovny bude definován prototyp funkce pro konverzi GIF89a na BMP. Dále bude definován typ záznamu udávající velikost souboru GIF a BMP v bytech. Knihovna musí umět převést statický barevný 8 bitový formát GIF. Převod 1-7 bitového formátu nemusí být implementován/podporován. Výstupní grafický soubor BMP nemusí být zkomprimován (RLE).

Datový typ záznamu o převodu bude mít tvar:

```
typedef struct{
    int64_t bmpSize;
    int64_t long gifSize;
} tGIF2BMP;
```

Prototyp funkce pro převod bude mít tvar:

```
/* gif2bmp – záznam o převodu
inputFile – vstupní soubor (GIF)
outputFile – výstupní soubor (BMP)
návrátová hodnota – 0 převod proběhl v pořádku, -1 při převodu došlo k
chybě, příp. nepodporuje daný formát GIF */
```

```
int gif2bmp(tGIF2BMP *gif2bmp, FILE *inputFile, FILE *outputFile);
```

V implementaci knihovny pak budou naimplementovány funkce pro dekompresi, a funkce pomocné (např. pro práci se slovníkem).

### **b) implementace aplikace:**

Aplikace se bude jmenovat *gif2bmp*, přičemž bude využívat implementovaných knihoven pro převod grafického formátu GIF na BMP s využitím kompresní metody LZW. Podporované parametry příkazového řádku jsou následující:

-i <ifile> název vstupního souboru <ifile>. Pokud parametr nebude zadán, bude se za vstupní soubor považovat standardní vstup (stdin).

-o <ofile> název výstupního souboru <ofile>. Pokud parametr nebude zadán, bude se za vstupní soubor považovat standardní výstup (stdout).

-l <logfile> název souboru výstupní zprávy <logfile>. Pokud parametr nebude zadán, tak výpis zprávy bude ignorován.  
-h vypíše nápovědu na standardní výstup a ukončí se.

Poznámka: Pro zpracování parametru příkazové řádky je výhodné využít funkce getopt (resp. getopt\_long), jejíž rozhraní je definováno v unistd.h (resp. getopt.h).

Výstupní zpráva bude obsahovat login, původní velikost souboru GIF (v bytech) a novou velikost BMP (v bytech), a to ve formátu dle přiloženého příkladu. Příklad zprávy aplikace autora Pepi Zdepa vypadá následovně:

```
login = xzdepa97
uncodedSize = 16384
codedSize = 14937
```

### **c) komentáře:**

Zdrojový kód komentujte tak, aby nebylo potřeba dodatečně projekt obhajovat. Komentujte zejména parametry a činnost funkcí, datové typy, lokální proměnné (pokud to nebude z názvu přímo vyplývat). Všechny zdrojové soubory budou obsahovat hlavičku, ve které bude jméno, příjmení a login autora. Dále bude v hlavičce datum vytvoření, název a stručný popis souboru.

### **d) kompilátor a prostředí:**

Knihovnu a výslednou musí být možné přeložit na serveru merlin, a to ve zde dostupné aktuální verzi kompilátoru GCC. Průběh překlada aplikace a jejího sestavování proběhne v prostředí GNU Make. K tomuto účelu bude sloužit soubor Makefile, k jehož interpretaci dojde zadáním příkazu make bez parametrů. Testování funkčnosti aplikace bude probíhat zásadně na serveru merlin.

### **e) dokumentace:**

Pro knihovnu a aplikaci napište stručnou dokumentaci. Součástí dokumentace bude stručný popis algoritmu. Rozsah dokumentace by neměl přesáhnout 2 strany A4 (= jeden list A4). Formát dokumentace bude PDF.

### **f) odevzdává se:**

Archív formátu ZIP, který se bude jmenovat kko.proj3."login".zip (např. pro Pepu Zdepa kko.proj3.xzdepa97.zip), bude obsahovat adresář kko.proj3."login" (např. kko.proj3.xzdepa97) s následujícím obsahem:

- gif2bmp.h – rozhraní knihovny
- gif2bmp.c – implementace knihovny
- main.c – aplikace
- Makefile – definice způsobu kompilace programem make
- gif2bmp.pdf – dokumentace

Pokud to bude z pohledu řešení účelné (lepší strukturování zdrojových kódů, modulárnost aplikace), je samozřejmě možné vytvořit i další soubory (různé pomocné knihovny, hlavičkové soubory, atd) kromě těchto požadovaných. Nezapomeňte je však umístit do odevzdávaného archívu.

### **Tipy a rady na závěr:**

- Důležitou vlastností je také přenositelnost. Pro různé platformy může být velikost typů int, long různá. Těmto problémům se vyhneme použitím předdefinovaných typů int8\_t, int16\_t, int32\_t, u\_int8\_t apod., které jsou definovány v knihovně sys/types.h.
- Pro ladění potíží s pamětí (Segmentation fault apod.) doporučuji použít nástroj valgrind, který se používá na aplikaci zkompilevanou pomocí make debug (příp. gcc -g -ggdb3).

Doporučují použít i když na první pohled je vše v pořádku, chybná práce s pamětí se může projevit výpadkem paměti až třeba u jiné platformy.

- V licenci GPL existuje také pro účely ladění pohodlný debugger *DDD* (DataDisplayDebugger), který je grafickou nadstavbou nad *gdb*. Opět je vhodné kompilovat kód pomocí *make debug*.
- Pro zpracování parametru příkazové řádky je výhodné využít funkce *getopt* (resp. *getopt\_long*), jejíž rozhraní je definováno v *unistd.h* (resp. *getopt.h*).
- Pro implementaci některých datových struktur a operací s nimi může být výhodné použít v aplikaci knihovnu STL (C++ Standard Template Library).

### **Doporučená literatura:**

[1] Salomon, D.: “Data Compression, The Complete Reference,” NY, USA, Springer, 2000, ISBN-0-387-95045-1

[2] Učební texty do předmětu Kódování a komprese dat, Brno, FIT VUT, 2006

[3] Sayood, K.: “Lossless Compression Handbook”, San Diego, CA, USA, Elsevier Press 2003, ISBN 0-12-620861-1

[4] Sayood, K.: “Introduction to Data Compression”, San Francisco, CA, USA, Elsevier Press 2006, ISBN -0-12-620862-X

[5] Murray, J. D., vanRyper W.: “Encyklopedie grafických formátů,” Praha, ČR, Computer Press 1997, ISBN 80-7226-033-2.

[6] Tišnovský, P.: Seriál článků “Grafický formát GIF”. Odkaz na URL:  
<http://www.root.cz/serialy/graficky-format-gif/>

[7] "GIF". Odkaz na URL: <http://cs.wikipedia.org/wiki/GIF>