

Semestrální úloha 36PJC

Programovací jazyk C

Téma úlohy: **Výpis EXIF informací z JPG souborů**

Vypracoval: **Michal Trs**

3. ročník, obor Výpočetní technika, K336 FEL ČVUT,
Karlovo nám. 13, 121 35 Praha 2

1. Zadání

Napište program, který ze zadaného JPG souborů přečte část s EXIF informací a zobrazí ji v textové formě na standardním výstupu.

2. Popis EXIFu

Jedná se o parsování binárního souboru. Podle specifikace Exif 2.2 Každý JPEG soubor začíná značkou SOI (má hodnotu FFD8) a je následován značkou APP1. Ta může mít hodnotu FFE1 jedná-li se o Exif nebo FFE0 a pak je to hlavička JFIF, nebo úplně něco jiného. Ihned za ním následují data APP1.

Jelikož podle zadání má program číst Exif informace, neřešil jsem JFIF a pouze je přeskočil a hledal FFE1. Nyní popíšu strukturu APP1.

Za značkou FFE1 jsou 2B znamenající velikost záznamu. Za velikostí následuje Exif a TIFF hlavička. Exif hlavička je dlouhá 6B a obsahuje řetězec „Exif“ následovaný dvěmi nulami. TIFF hlavička je dlouhá 8B.

První 2B znamenají jak jsou data v souboru uložena. Pokud mají hodnotu 0x4949="I I" (Intel) je nižší byte uloženy na nižší adrese (little endian), Pokud je hodnota 0x4d4d="MM" (Motorola) jedná se o Big endian. Dále je zapsána 2B hodnota 2A00 a za ní 4B offset na první položku IFD (Image File Directory). Popis hlavičky je shrnut v následující tabulce.

Big / little endian	Značka TAGu	Offset na první IFD
II nebo MM	2A00	nejčastěji 0x00000008

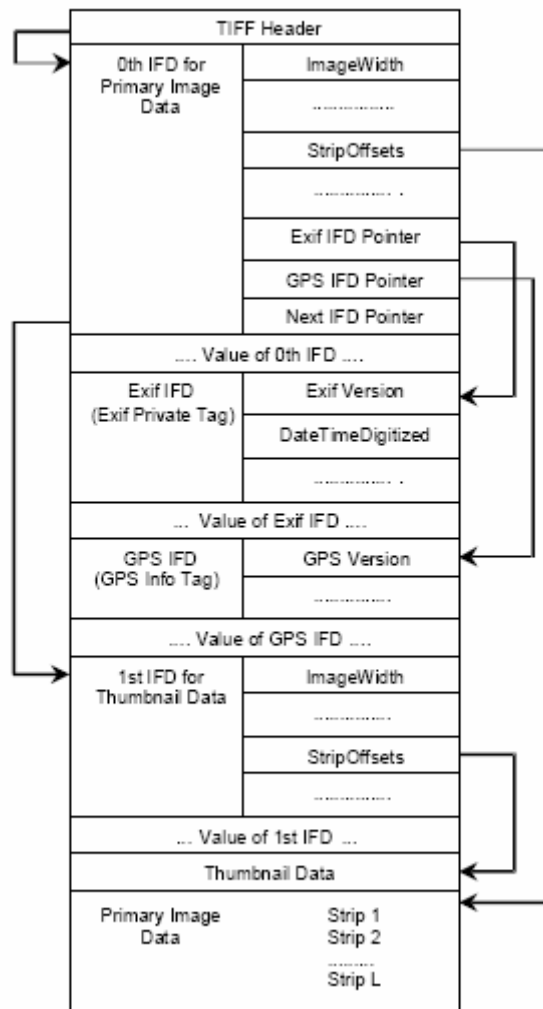
IFD položka

Za hlavičkou TIFF je první IFD adresář. První 2B je číslo udávající počet položek. Každá položka má 12B. Za poslední položkou jsou 4B udávající offset na další IFD adresář.

Položka se skládá ze 4 částí.

- 1) první 2B je číslo tagu
- 2) další 2B udávají formát v jakém je uložena hodnota
- 3) 4B udávají počet komponent
- 4) Samotná hodnota, nebo pokud je větší než 4B obsahuje offset

Na následujícím obrázku je vidět uspořádání informací za TIFF hlavičkou.



3. Popis implementace

Program je rozdělen do 4 .c souborů a 3 hlavičkových .h souborů. V data.h jsou definovány struktury pro ukládání názvů tagů a případně jejich hodnot. V data.c jsou soustředěny téměř všechny textové řetězce vypisovány programem. Pokud např. bude požadavek předělat výpis v jiném jazyce stačí změnit tento soubor. V pfunc.h a pfunc.c jsou funkce starající se o výpis informací převážně z data.c. Jsou zde jednoduché funkce pro výpis jednotlivých datových typů, a několik komplexnějších funkcí, které zde popíšu.

`int ErrMsg(int errCode);` – Podle chybového kódu vypíše chybovou hlášku na standardní chybový výstup. Vrací chybový kód s kterým byla funkce volána.

`int printIFDname(unsigned int tagID, IFDtype typ);` – Pokud bylo nalezeno tagID vytiskne se jemu odpovídající jméno a vrátí se 0. Pokud tagID nebylo nalezeno, nic se netiskne a vrátí se hodnota různá od 0.

```
int printIFDvalue(unsigned int tagID, unsigned int df,  
unsigned long int coc, unsigned long int index, int decode);  
– Vypíše hodnotu tagu. Pokud není vypnuto dekodování je v spec[]  
vyhledán slovní význam hodnoty.
```

```
void showHelp(void); – zobrazí nápovědu. Samotná nápověda je  
řetězec definován v *help
```

V další souboru jsou definovány proměnné a jednoduché funkce pro práci s pamětí. Při parsování po zjištění velikosti TIFF části, je celá načtena do paměti a parsování probíhá nadále v paměti.

Hlavní soubor main.c obsahuje jak samotnou funkci main, tak funkce pro samotné parsování a nějaké pomocné funkce.

Deklarace funkcí zajišťující celé parsování:

```
int readAPP1header(void) – načte APP1 hlavičku, na základě této  
funkce je TIFF část načtena do paměti.
```

```
int readTIFFheader(void) – nastaví Little / big endian
```

```
int readIFD(IFDtype typ) – podle IFDtype přečte celou strukturu IFD,  
volá další funkce z main.c a pfunc.h
```

4. Popis programu

Program je přenositelný mezi windows a unix systémy. Ke kompilaci je využít makefile.

Pokud je program spuštěn bez parametrů nebo s jedním parametrem a to -h nebo --help bez dalších argumentů je vypsána nápověda a program se ukončí.

Spuštění:

```
exifr [-d] file.jpg
```

kde: -d nepovinný parametr, vypne dekodování hodnoty tagů na slovní význam

file.jpg JPEG soubor obsahující EXIF informace, pokud je soubor neobsahuje, nebo je nějakým způsobem poničen, program končí chybovou hláškou a chybovým kódem.

Chybové kódy a chybové hlášky:

- 1 Can't open file
- 2 Can't read the file!
- 3 File hasn't JPEG format!
- 4 EXIF information isn't available!
- 5 Error in file!
- 7 Unexpected end of file
- 10 Can't close file"
- 15 unknown parametr. type -h for help
- 255 Unknown error

5. Závěr

Program splňuje zadání. Lze jej snadno rozšířit o výpis jiných IFD adresářů (např. GPS tagů).

6. Literatura

<http://www.exif.org/>

<http://www.exif.org/Exif2-2.PDF>